Anno VIII - N. 64 - 25 maggio 1989
Sped. Abb. Post. Gr. III/70 CR -Distr.: MePe

Commodore COMPUTER CLUB

64 L. 5.000

La rivista degli utenti di sistemi Commodore

LA GUERRA DEI VIRUS

Ed io infetto il tuo vaccino

C 64/128

- **Equazioni differenziali**
- **Gestione dello scrolling**

C 128

Lo Z 80 senza CPM

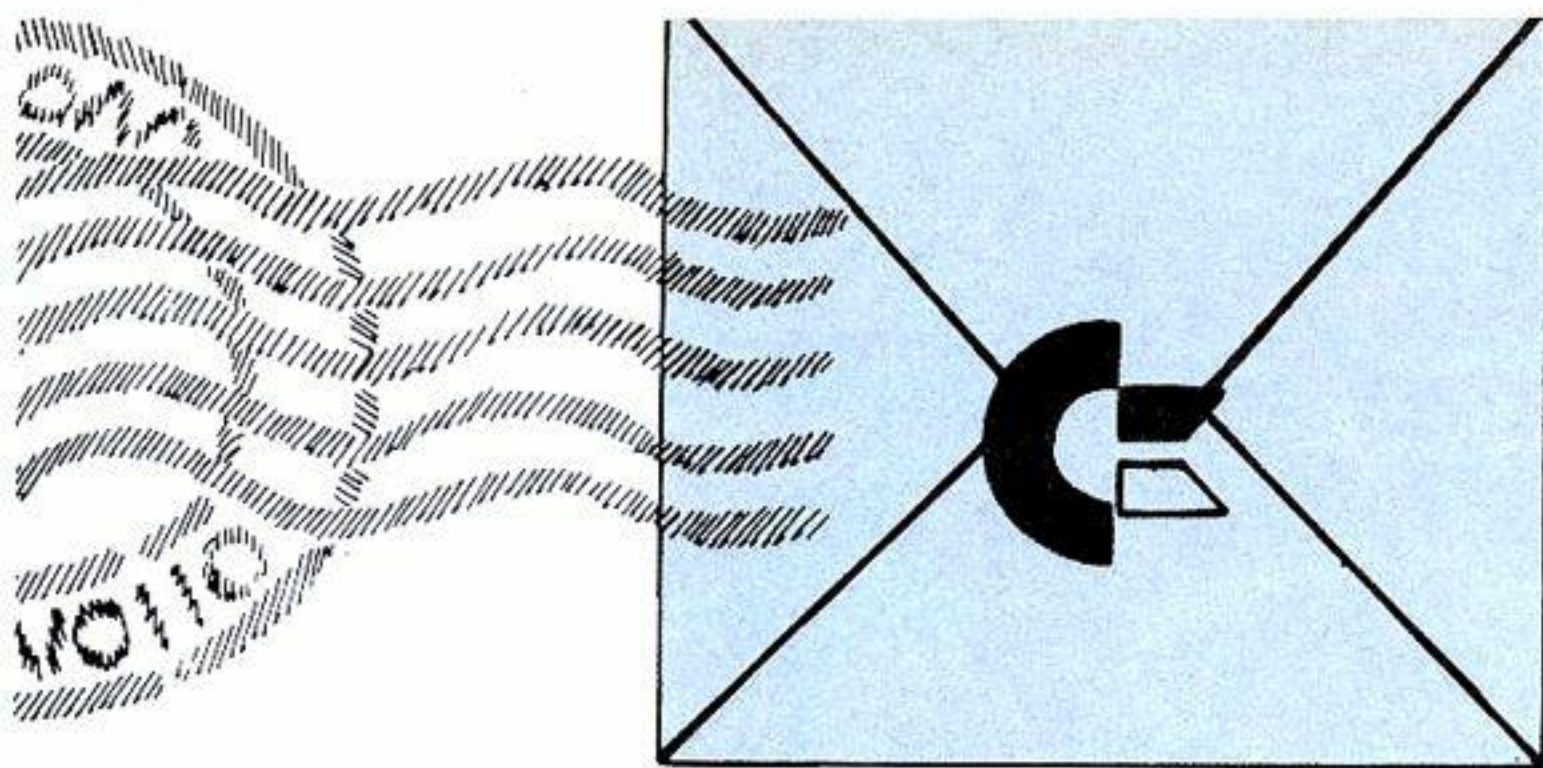
Amiga

Un pantografo col mouse

Turbo Pascal 64
Come usarlo meglio

Sommario

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Angelo Bianchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma, Franco Rodella, Stefano Simonelli
Grafica: Arturo Ciaglia
Direzione, pubblicità: via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310 - **Redazione:** Tel. 02/5249211
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Angelo Ricupero - Via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Arretrati e software: Lucia Dominioni (Tel. 02/8467348/9 - viale Famagosta 75 - 20142 Milano)
Tariffe: prezzo per copia L. 5.000. Abbonamento annuo (11 fascicoli) L. 50.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 90.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib.:** MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age -Energy Manager - Hospital Management - Jonathan - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - TuttoGatto - Videoteca - VR Videoregistrare

la vostra posta

VELOCITA'

□ Ho visto in commercio un microprocessore 68000 che può operare a 16 Mhz. Che vantaggi avrei sostituendolo a quello presente sul mio Amiga?

(Luca Vialle - Napoli)

• Nessuno. Il chip di cui parla il nostro lettore, infatti, per "viaggiare" a 16 Mhz ha bisogno della circuiteria complementare (l'oscillatore, tra l'altro) che gli consenta di raggiungere tali velocità.

Per quanto riguarda un circuito elettronico, infatti, è necessario che tutti i componenti della piastra abbiano eguali caratteristiche. Non avrebbe senso, in altre parole, aumentare la velocità operativa del solo microprocessore se, ad esempio, le memorie Ram non sono in grado di operare alle stesse frequenze. Una memoria RAM ha bisogno di un certo intervallo di tempo per cambiar stato e potrebbero sorger problemi se questo fosse maggiore del tempo impiegato dal micro per impartire i vari ordini.

Altri circuiti integrati, inoltre, richiedono tempo per svolgere le proprie operazioni e potrebbero presentare problemi analoghi.

Insomma, per andare più veloci non è sufficiente mettere il motore di una Ferrari in una 500; bisogna adattare i freni, le gomme, il sistema di raffreddamento e tante di quelle "cose" che spesso si è costretti ad abbandonare il progetto.

BRUTTI SCHERZI

□ Cercando di caricare Amigabasic dal disco Extras, come ero abituato a fare di solito, è comparsa una segnalazione di errore relativa ad un Read error. Da allora non riesco più ad utilizzare il dischetto in questione. Che cosa è potuto accadere?

(Alessandro Giovinazzo - Vibomarina)

• Molto probabilmente il dischetto si è rovinato "fisicamente" e, se questo è il caso, non c'è niente da fare, nemmeno operando con il programma Diskdoctor, presente su Workbench. Dirò di più: se il difetto è di tipo fisico, non sarà nemmeno possibile formattare nuovamente il disco.

Anche noi abbiamo notato una eccessiva delicatezza dei dischetti trattati da Amiga e dal momento che usiamo dischi di identica qualità anche con altri nostri computer Ms-Dos (che non hanno mai presentato problemi di sorta) incominciamo a pensare che, in effetti, i floppy da usare su Amiga devono essere di qualità "superiore", come suggerito sullo stesso manuale del computer.

Approfittiamo della lettera pervenuta sia per ricordare che è necessario lavorare SEMPRE su copie (mai sugli originali!), sia

AIUTATECI A SERVIRVI MEGLIO

Spesso alcuni lettori porgono quesiti le cui risposte sono già state esplicitamente pubblicate (in occasione di risposte ad analoghe domande) oppure sono contenute in articoli presenti nei fascicoli in loro possesso.

Si ricorda ai lettori che non ci è possibile rispondere privatamente, nemmeno se si acclude l'affrancatura per la risposta.

Per accelerare il servizio, ricordate di indicare sempre la data di spedizione dal momento che questa costituisce diritto di precedenza.

A tutt'oggi (con la pubblicazione del presente fascicolo) risulta evasa l'intera corrispondenza pervenuta in Redazione entro la fine del mese appena trascorso.

Non ci è possibile dare consigli su acquisti da effettuare nè esprimere pareri estranei alla logica della correttezza professionale e commerciale.

ABBIAMO GIA' RISPOSTO

I seguenti lettori possono trovare adeguata risposta ai loro quesiti esaminando con cura i libretti di istruzione delle apparecchiature, o del software, in loro possesso, gli arretrati di "Commodore Computer Club" che posseggono o i nostri fascicoli nel frattempo apparsi in edicola.

Silvano Berardi - S. Ippolito; Francesco Bernardi - Genova; Marco Campinoti - Venturini; Gianluca Ciavaglia - G. Tadino; Franco Costantini - Cerveteri; Giuliano Galante - Palermo; Gianluca Maiolini - Ancona; Domenico Misceo - Bari; Paolo Parsella - Fondi; Daniele Rancati - P.to Torres; Paolo Redaelli - Desio; Marco Sampietro; Emilio Soresi - Partinico; Francesco Soluri - Gimigliano; Marco Stefanucci - Roma; Massimo - Rm;

RISPONDEREMO TRA BREVE

I lettori qui di seguito nominati troveranno opportuni chiarimenti in fascicoli di imminente pubblicazione.

Alessandro Ambrosini - Genova; Maurizio Castelli - Villaminozzo; Stefano Crimì - Caltanissetta; Giuseppe Doni - Lido di V.; Enrico Fantinel - Seren del Grappa; Fabio Lisco - Bari; Manganiello Gianluca - A. Irpino;

Spett.le
C.C.C.
Via Mosè
OPERA MI

per chiedere il parere di altri utenti Amiga che ritengono di avere analoghi problemi con il drive.

Ma come si fa a lavorare su copie se si utilizzano programmi originali protetti? Bè, non possiamo rispondere proprio a tutto...

ACCESSORI PER AMIGA

☐ Vorrei sapere come (e dove) procurarmi un drive 5.25, il linguaggio Turbo Pascal ed il volume "Amiga Rom Kernel" per sfruttare al meglio il mio Amiga.

(Paolo Selvini - Como)

• Dovrebbe esistere anche il drive da 5.25, ma finora non abbiamo avuto modo di provarlo (a proposito, a che ti servirebbe?). Per ciò che riguarda il Turbo Pascal, ho avuto modo di vedere, tempo fa, una confezione del software (in inglese) per tale linguaggio. Non so se in Italia c'è qualche importatore che abbia voglia di commercializzare (SERIAMENTE) il versatile linguaggio, ma dubito che ci sia qualcuno disposto a distogliere energie dal ben più florido mercato dei videogame.

Per ciò che riguarda i testi inglesi su Amiga (tra cui quello menzionato) è necessario recarsi presso librerie molto, molto specializzate: noi l'abbiamo trovato presso la libreria Hoepli di Milano. Attenzione, però, alla versione delle Rom descritte: sono in giro le vecchie edizioni dei volumi che parlano delle Rom 1.1.

LA VOCE DEI LETTORI

OPINIONI

La rubrica "La voce dei lettori" è nata per dare spazio (e sfogo...) ai lettori che, pur non avendo nulla di particolare da chiedere, intendono far sentire la propria voce, chiedere solidarietà ad altri utenti o, semplicemente, riportare osservazioni ritenute utili o importanti. Per questioni di spazio le lettere sono spesso condensate: vengono poi "censurate" le osservazioni ed i termini troppo "forti" che possano arrecare offesa ai lettori, agli inserzionisti e... a noi stessi. In calce viene riportata la sola iniziale del cognome del lettore (salvo casi particolari), per fare in modo che questi possa sentirsi ancor più libero di esprimere la propria opinione. Coloro che desiderano il minor numero di "tagli" possibile alle loro missive sono pregati di inviarle su DISCO, usando il word processor Easy Script.

PIU' MEMORIA

☐ Impartendo Poke 56, 255 (sul C/64) e digitando, successivamente, Print Fre(0), il computer risponde che sono disponibili 63218 byte. Come è possibile una cosa del genere?
(Luca Vignale - Brandizzo)

• E infatti non è possibile. La locazione 56, all'accensione del C/64, contiene il valore 160 che, moltiplicato per 256, fornisce il risultato 40960 che indica l'ultima locazione Ram disponibile da Basic.

Modificando il contenuto della locazione 56 (mediante Poke) si inganna il computer facendogli credere che, in effetti, l'ultima locazione disponibile è proprio... l'ultima possibile (256*256 = 65535). La gestione delle varie elaborazioni, però, risulta alterata ed è facile madare in Tilt il computer.

Nonostante la modifica, infatti, è ancora possibile assegnare valori a variabili numeriche (A=23: C=18 e così via), ma accadono cose strane tentando di trattare stringhe. Provando a digitare...

A$ = "Pippo": Print A$

...vedrai apparire caratteri privi di senso. Allo stesso modo, tentando di caricare programmi troppo lunghi, possono verificarsi malfunzionamenti di vario tipo.

E' proprio vero: nulla si crea e nulla si distrugge. Alla rigida regola non sfuggono nemmeno le Ram...

DPS 1101

☐ Possiedo la stampante a margherita Commodore DPS - 1101 e non riesco a far apparire i caratteri maiuscoli con nessuna delle due routine di hard copy pubblicate sui numeri 51 e 53 di C.C.C.
(Cristoforo Paiano - Lecce)

LA VOCE DEI LETTORI

DALLA PARTE DEL PLUS 4

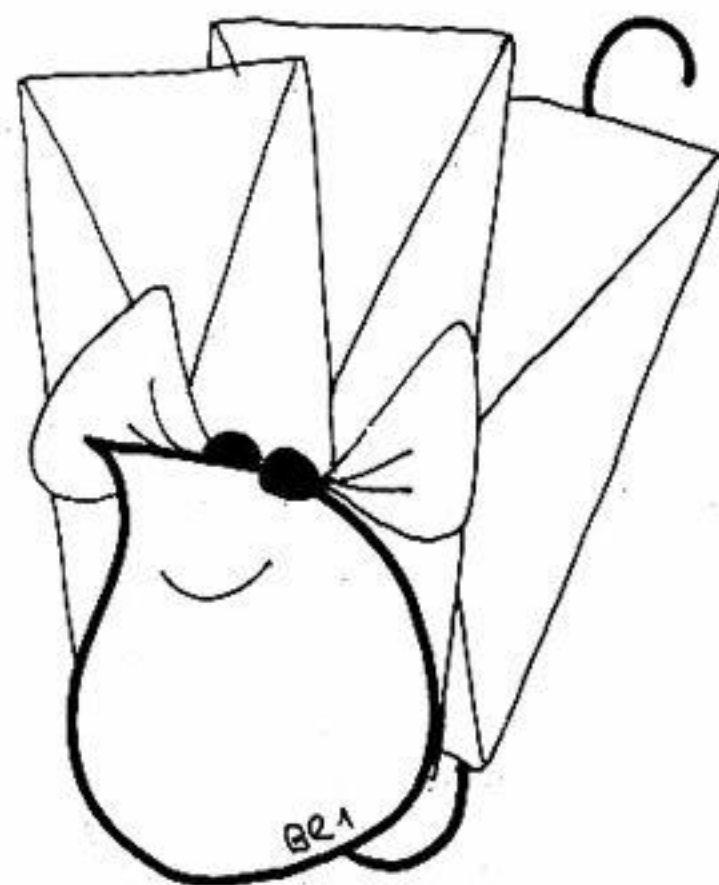
Quale possessore di Plus 4 mi associo totalmente a quanto detto dai sigg. Roberto N. di Bologna, Nino P. di Cagliari, Alessandro S. di C. Veneto aggiungendo che la Commodore ha perso un'occasione di sviluppo nel trascurare e (direi addirittura) boicottare uno dei suoi più riusciti minicomputer, il Plus 4 appunto.

Il suo uso sarebbe stato idoneo sia per i giochi che per programmi applicativi e didattici più seri, cui avrebbe fatto senz'altro riscontro un'utenza più eterogenea e numerosa; da non dimenticare i quattro programmi incorporati nella macchina.

Invece noi "figliastri" Commodore siamo costretti a rinunciare alle realizzazioni, in gran parte indirette, effettuate per il C/64 e che invece potevano esser fatte direttamente sul Plus 4 grazie al suo potente Basic.

Se deciderò per un sistema superiore, questo non sarà di marca Commodore, come già ha deciso il citato Sig. Roberto N. di Bologna.

Grazie per l'accoglienza ed un saluto ai 16-sti e Plusquattristi: allegri ragazzi, siamo in pochi ma non rincretiniti.

Vincenzo M. - Avezzano

DALLA PARTE DEL C/64

LA VOCE DEI LETTORI

Sono da poco lettore di C.C.C. e, anche se non molto esperto, ho il C/64 e voglio vivamente contestare ciò che Nino P. di Cagliari ha scritto sul N. 62. Ritengo che il C/64 sia stato il primo computer costruito un po' decentemente! Capisco che altri elaboratori siano più perfezionati del C/64, ma non certo il C/16 o il Plus 4 (scarsa memoria e poche prestazioni). Penso che il C/64 abbia una bella grafica per i videogames, ed anche un ottimo Basic.

Cristiano L. Torino

Risposta a Nino P. di Cagliari. Ritengo doveroso dirle che prima di parlare a vanvera sarebbe meglio che riflettesse sulle cose dette e si pentisse. Innanzitutto sono in disaccordo con le sue affermazioni, sia da un punto di vista educativo che ricreativo. E' vero che il C/64 ha un Basic molto povero, scarno e privo di specifiche istruzioni per gestire la grafica, il suono e così via, ma, proprio per questo motivo, è uno strumento molto valido se riferito alla programmazione. Mi spiego meglio con un esempio volutamente banale: nei Basic più evoluti è presente l'istruzione Swap, che consente lo scambio tra due variabili (swap a, b). Con il Basic del C/64 si è costretti ad ottenere lo stesso risultato con tre istruzioni (d=a: a=b: b=d) e si è quindi obbligati a mettere in moto la materia grigia del cervello per cercare soluzioni alternative. Modestia a parte, ritengo di essere piuttosto bravo nella programmazione, ma se ho raggiunto tale preparazione lo devo esclusivamente al C/64 che, per l'appunto, mi costringeva a ricercare nuove alternative per rendere più efficiente ogni programma.

Oggi come oggi ringrazio la Systems Editoriale per i prodotti editi (Ms-Dos, Gw-Basic, Turbo Pascal) che aumentano la capacità del piccolo C/64 e lo proiettano (anche se modestamente) nel mondo dei veri Personal Computer. Ricordo infine al signor Nino P. che il C/64 è un home computer e, come tale, deve servire anche per il tempo libero: ci si può quindi divertire, utilizzando i videogame, senza avere scrupoli sulla coscienza.

Claudio

• Le due routine pubblicate sono idonee per ottenere hard copy (in modo testo per C/64 e per C/128 in modalità 40 e/o 80 colonne) con stampanti 803 compatibili. La DPS - 1101 non è 803 compatibile ma, trattandosi di modo testo, dovrebbe funzionare egualmente bene dal momento che entrambe le routine impongono il suffisso ",7" (indirizzo secondario) per attivare la modalità maiuscolo / minuscolo.

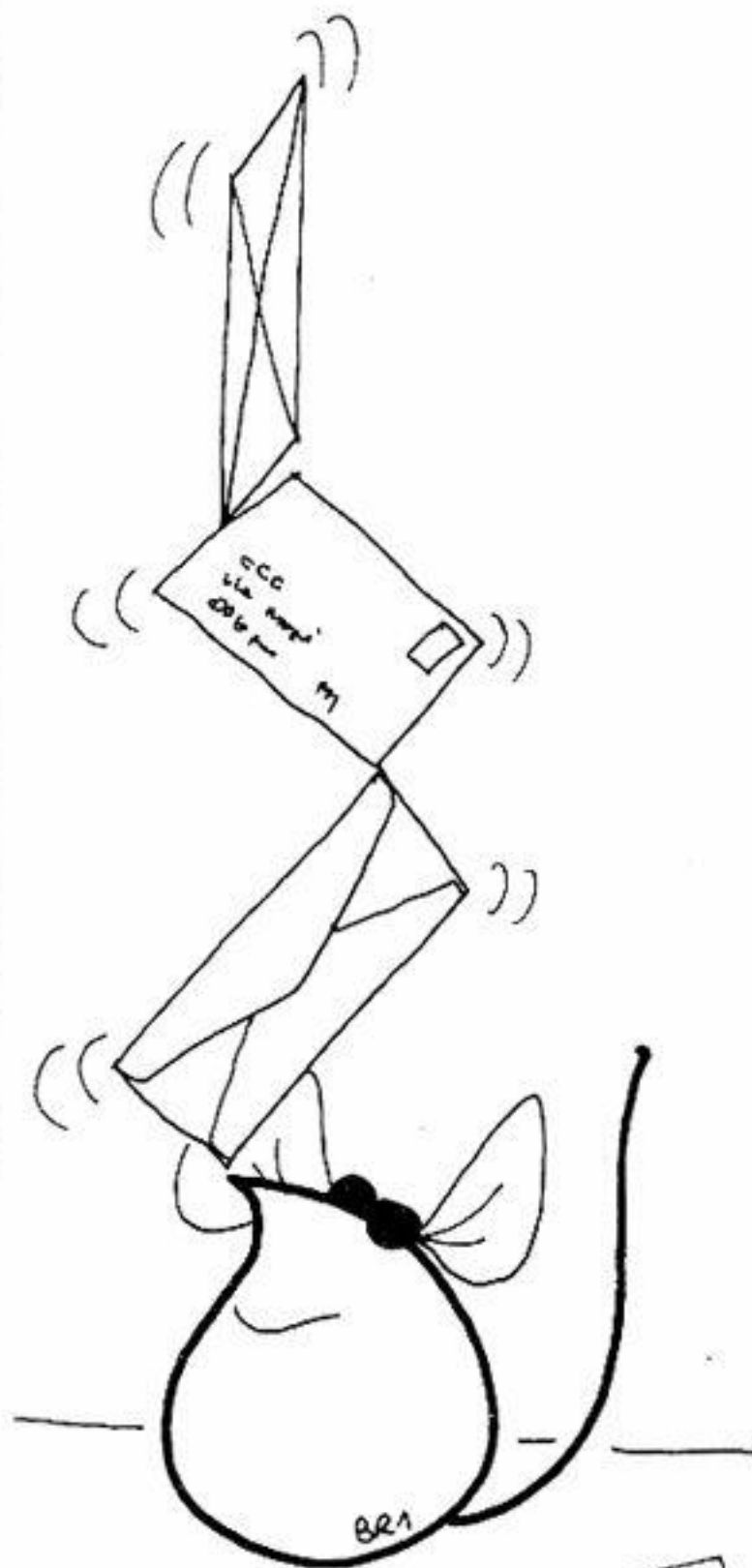
Purtroppo, non disponendo della stampante in oggetto, non possiamo effettuare le prove pratiche per fornire una corretta informazione al riguardo.

Se qualcuno ha risolto il problema del nostro lettore, si faccia avanti!

POKE PERICOLOSE

☐ In un videogame scritto in Basic ho notato la presenza di Poke 809, 255. Che cosa rappresenta la locazione 809 dal momento che, se cancello l'istruzione in oggetto, il programma si blocca?
(Fabio Tiezzi - Grosseto)

• La coppia di locazioni 808 e 809 rappresenta un vettore del Kernal, vale a dire l'indirizzo a partire dal quale inizia la routine Stop. Alterando i suoi valori si rischia, come appunto hai notato, di mandare in tilt il sistema.

Progettando opportunamente tale vettore è possibile escogitare sistemi di protezione che possono agire in moltissimi modi diversi tra loro. Il più banale di questi consiste nell'impedire il funzionamento del tasto Run/Stop o di far ripartire il programma provando a premere Run/Stop e Re-

MAESTRI, SOLDI E COMPUTER

LA VOCE DEI LETTORI

L'uso del registratore è (parole vostre) "paleoinformatica". Perfettamente d'accordo in linea generale, ma in una scuola in cui i computer entrano solo se li comprano i maestri, forse ci sono altre considerazioni da fare. Personalmente posseggo ben tre sistemi equipaggiati di drive, ma anche di registratore; anzi, il mio primo sistema personale era sfornito di registratore. L'ho dovuto acquistare perchè i miei alunni (scuola elementare) posseggono, a casa loro, solo questa periferica ed è mio dovere tener conto di tale particolare. Del resto, usando accortamente il datassette, si possono realizzare bei programmi senza far salti mortali o tentare procedure delicate e "pericolose". E dal momento che siamo in argomento, non riesco a capire perchè mai nelle scuole italiane si sia aperta la porta ai sistemi Ms-Dos che, rapportati ai ben più economici e diffusi C/64, pochissimi studenti posseggono. Tra i miei allievi che possiedono un computer, ad esempio, 6 hanno il C/64, 1 un Msx e 1 un Prodest. Ma questi ultimi sono dei "pentiti"...

Mario C. Vercelli

store. Ogni programma può avere una tecnica diversa ed è quindi impossibile individuare il modo di operare del programma in tuo possesso.

RIGHE LUNGH(ISSIM)E

□ In alcuni listati ho notato la presenza di righe Basic contenenti un numero di caratteri ben superiore agli 80 consentiti. Come posso realizzare anch'io una cosa del genere?
(Gioacchino Vitrano - Ciavolotto)

• E' possibile ottenere righe Basic lunghe molto più di 80 caratteri: è infatti sufficiente, in fase di scrittura del programma stesso, ricorrere alle abbreviazioni che sono riportate sul libretto di istruzioni del computer. Ad esempio, invece di...

```
10 Print "Prova"
```

...è sufficiente digitare il punto di domanda (?)...

```
10 ? "Prova"
```

Ad un successivo comando di List, tuttavia, il punto di domanda viene sostituito automaticamente dalla parola - comando "Print". E' quindi intuitivo che, riempendo un'intera riga di punti di domanda e di doppi punti...

```
10 ?: ?: ?: ?: ?
```

...è facile superare gli 80 caratteri.

Attenzione, però: quando, dopo il list, viene visualizzata la riga, non sarà possibile effettuare modifiche, al suo interno, se la lunghezza totale supera gli 80 caratteri (cioè le fatidiche due righe di schermo). Il sistema di editazione del C/64, infatti, considera valida solo la riga "logica" Basic lunga, al massimo, due righe video. La parte "eccedente", quindi, pur se perfettamente "accettata" (e funzionante), viene totalmente ignorata nel caso ci si posizioni sopra con il cursore e si prema il tasto Return.

DISABILITARE TASTI

□ Come è possibile disabilitare i tasti del controllo cursore?
(Giancarlo Fagone - Palermo)

• Non capisco quale possa essere la pratica utilità di un simile controllo. Tieni presente, comunque, che vi possono essere due modi per farlo.

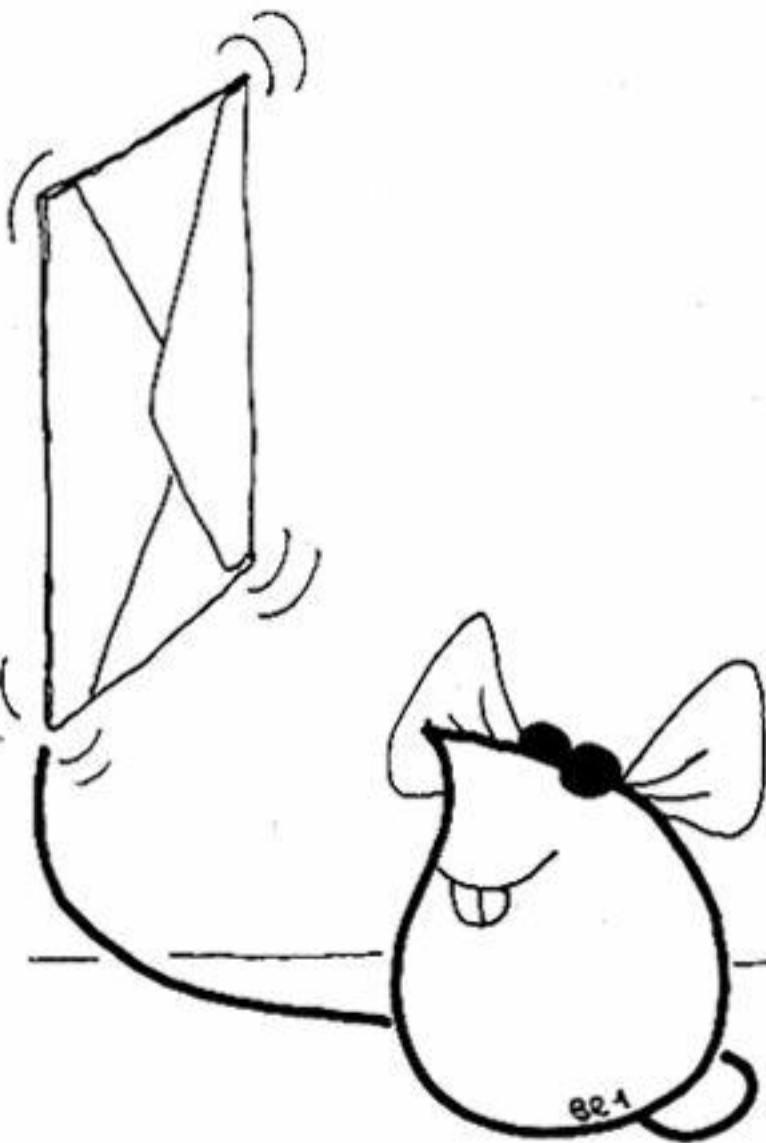
Il primo di questi, che è anche il più complesso, impedisce il riconoscimento di eventuali pressioni dei tasti interessati grazie alla manipolazione dell'interrupt. In pratica si tratta di elaborare una routine in linguaggio macchina (da scrivere appositamente) che, interagendo con lo stesso sistema operativo, è in grado di stabilire il tasto premuto e, in casi particolari, ignorare l'evento (pressione dei tasti di controllo cursore) o procedere in modo diverso dal consueto.

La tecnica, molto particolare, può avere risvolti negativi nel caso la routine di cui sopra rimanga attiva durante il funzionamento di altri programmi in linguaggio macchina (come videogame o professionali) che manipolano anch'essi l'interrupt di sistema.

Il secondo metodo, ben più semplice del primo, funziona in normalissimi programmi Basic e fa parte della miriade delle subroutine di "Input controllato".

Una soluzione può essere la seguente:

```
100 print "prova a digitare"
110 get a$: if a$="" then 110
120 a=asc(a$)
130 if a=17 or a=145 or a=29 or a=157
then goto 110
140 print a$;: goto 110
```

I codici Ascii 17, 145, 29, 157 si riferiscono, infatti, ai quattro tasti cursore di un C/64.

Per individuare il codice di un qualsiasi tasto (che può cambiare a seconda del computer usato) un programma idoneo può essere il seguente:

```
110 get a$: if a$="" then 110
115 print "codice =";
120 print asc(a$): goto 110
```

MPS-803 PER AMIGA

□ Ho sentito dire che esiste un'interfaccia, o qualcosa del genere, che permetterebbe di collegare all'Amiga la stampante MPS-803. Di che si tratta?
(Anonimo privo di nome)

• Qualche numero fa scrissi che, pur avendo sentito parlare di accessori del genere (che non rappresentano nulla di trascendentale e non dovrebbero, di conseguenza, creare problemi) non me la sentivo di esprimere un parere su oggetti non provati personalmente. E ho fatto bene: un nostro valido collaboratore, infatti, ha tentato di collegare ben due modelli diversi di interfaccia tra il suo Amiga e Mps-803 (rigorosamente originale Commodore) ma non hanno voluto saperne di funzionare.

I negozianti che gli avevano venduto le interfacce sono stati costretti ad ammettere che non era il primo caso e che l'interfaccia non sembra andar d'accordo con alcuni esemplari di Amiga.

Il consiglio è il solito: avere precise garanzie di funzionamento (PRIMA di acquistare qualsiasi apparecchio) oppure comprarlo con l'esplicito accordo di restituzione, nel caso non dovesse funzionare correttamente.

PROGRAMMI SU ORDINAZIONE

□ Vorrei veder pubblicati programmi per la soluzione di problemi riguardanti l'analisi matematica (serie numeriche, integrali, eccetera).
(Salvatore Incardona - Palermo)

• I programmi citati dal nostro lettore (studente universitario di Ingegneria) potrebbero interessare solo una ristretta cerchia di utenti e dubito che possano essere scritti e pubblicati. Di solito, infatti, siamo abituati a "fermarci" agli argomenti oggetti di studio delle scuole superiori e non ci spingiamo fino all'università.

Se qualche lettore, tuttavia, ha realizzato qualcosa del genere, è pregato di mettersi in contatto con la Redazione, in modo da valutare l'opportunità della pubblicazione del listato e relativo articolo.

PERIKOLO: POKE

□ Alterando il contenuto di alcune locazioni relative ad un programma Basic, capita di ottenere una linea a numerazione più elevata che precede una di numerazione più bassa. Come è possibile una cosa del genere?
(Roberto Corelli - Genova)

• Digitando il semplicissimo listatino che segue...

```
100 rem numerazione.
110 rem lista prima e dopo!
120 poke 2051, 255
130 list
```

...e impartendo il Run ci si accorge, con stupore, che la riga 100 non c'è più, ma è stata sostituita da una riga 255 che, per di più, è la prima della lista.

Il motivo è dovuto all'alterazione di uno dei due byte relativi alla numerazione della PRIMA riga Basic; l'altra locazione è la 2052 (prova ad eseguire Poke 2052, 255 e a visualizzare il listato!).

Il fenomeno è del tutto normale e la "mancata" automazione del giusto incolonnamento è dovuta esclusivamente al fatto che lo stesso automatismo viene attivato solo operando in Editing di Basic (in altra parole, digitando linee Basic e premendo il tasto Return).

Operando con le Poke, che agiscono in area programma, tale automatismo viene escluso e il computer, nel visualizzare il listato, non può rendersi conto di eventuali incongruenze, proprio perchè presume che le righe siano state già esaminate (ed incolonnate) nella precedente fase di editing.

La manipolazione introdotta dal nostro lettore, tuttavia, è utile per varie tecniche di protezione o di curiosità di varia natura. Provando, poi, ad alterare i puntatori alla successiva linea Basic si vedranno cose stranissime.

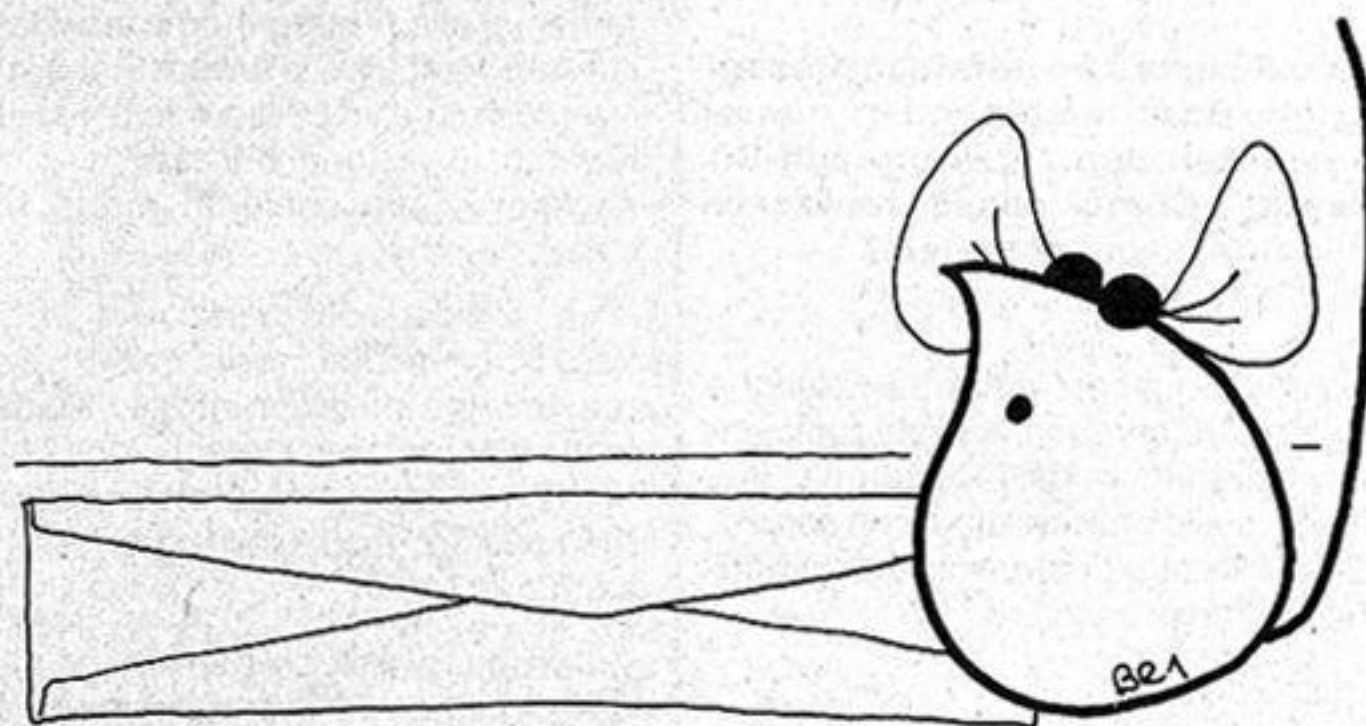
L'argomento, come avrai osservato, è piuttosto complesso e, del resto, affrontato più di una volta nei numeri precedenti di C.C.C. (come anche gli altri argomenti accennati nella lettera).

PROGRAMMI PERMANENTI

□ Vi sono alcuni computer che consentono di tenere in memoria un programma anche se l'apparecchio viene spento. E' possibile fare lo stesso con un C/64?
(Stefano Buonomo - Caiazzo)

• I computer portatili sono realizzati con componenti elettronici in grado di assorbi-

IN ATTESA DI GIUDIZIO

Rispondiamo brevemente, con un succinto giudizio, ai lettori che hanno proposto la pubblicazione di programmi ed articoli originali.

GESTIONE BORSA
(Luigi Allemandi - Luserna S.Giovanni)

I programmi inviati sono troppo lunghi per esser pubblicati, benchè risultino piuttosto curati e scritti con una tecnica di programmazione sufficientemente strutturata in gruppi di subroutine.

La notevole attenzione dedicata all'estetica di presentazione (incolonnamento messaggi, lampeggio degli stessi, colori di vario tipo) ha impedito di concentrarti sulle effettive difficoltà che un utente inesperto può incontrare usando il software in oggetto. Inoltre manca la possibilità di ritornare al menu principale in qualsiasi momento (mediante, magari, la pressione di tasto particolare) ed è seccante l'impossibilità di correggere i dati dopo averli inseriti. Manca infine (se non erro) un controllo su eventuali incongruenze di date inesistenti e non è possibile iniziare ad operare ad anno iniziato, ma solo a partire da gennaio. Inoltre il programma non consente di escludere le quotazioni di alcune giornate che potrebbero non interessare (vendita di tutte le azioni di un gruppo, mancanza di dati e così via). Le variabili usate sono troppe ed impediscono, a chi lo desiderasse, di intervenire sul programma stesso.

WORD PROCESSOR PER C/128
(Federico Alpi - Roma)

Il programma che hai inviato (ma perchè non hai preso preventivi accordi telefonici?) non può esser preso in considerazione per numerosi motivi che ora espongo e che sono validi per tutti i lettori-aspiranti-collaboratori della nostra rivista.

1- Non sempre (anzi, mai) conviene scrivere programmi appartenenti alla cosiddetta "triade" (word processor, data base, spreadsheet) perchè sono disponibili da tempo immemorabile versioni estremamente raffinate, sofisticate, sperimentate e, soprattutto, (ahinoi) piratate. Queste sono, di conseguenza, quasi gratis per tutti, e nessuno può avere interesse ad usare nuovi programmi. Per il C/128, poi, esistono numerosi W/p la cui potenza difficilmente può essere eguagliata (o superata), soprattutto se ci si limita al Basic.

2- Il numero di blocchi occupati dal programma inviato (ben 135!) ci impedisce di inserire il listato su Software Club, anche se a titolo gratuito.

3- Il programma non risulta molto ben strutturato e, se non bastasse, funziona sul C/128 in 80 colonne, modalità che non tutti gli utenti 128-isti usano spesso o volentieri.

4- La presenza di una password rende ancor meno versatile l'uso del software proposto.
5- Dulcis in fundo: anche se il programma fosse stato valido, non l'avrei preso in considerazione perchè, sul dischetto inviato, manca l'articolo esplicativo che, invece, ti sei limitato ad inviare su carta. Ripeto per l'ennesima volta che non possiamo assolutamente digitare i vostri articoli. Abbiamo a disposizione, infatti, solo il tempo di correggere (a video) quelli inviati su supporto magnetico che, dopo la correzione, vengono inviati direttamente (ed elettronicamente) in fotocomposizione (= macchina per la stampa).

Nonostante le note precedenti, devo riconoscere che la pazienza (e la bravura) dimostrate dal nostro lettore nella stesura del programma sono certamente lodevoli ed è un peccato che le indubbie capacità non vengano indirizzate verso lo studio del linguaggio macchina, per il quale vale senz'altro la pena dedicare tempo e risorse.

Per ciò che riguarda i programmi della triade, preciso che possono destare interesse listati brevissimi, che siano in grado di svolgere applicazioni originali e, in ogni caso, non svolte dai programmi commercializzati; altrimenti non ne vale la pena.

SCROLLING EDITOR
(Simone Balestra - Codigoro)

Il programma in l.m. è troppo lungo per la pubblicazione. Le spiegazioni per il funzionamento, inoltre, sono troppo scarne: non si riesce bene a capire quale sia l'utilità pratica del programma inviato. La prossima volta ti consiglio di telefonare PRIMA di inviare il materiale.

DAC MANAGER
(Davide Pagliara - Massafra)

Il programma è troppo lungo per la pubblicazione e, tra l'altro, le numerose routine da cui è formato farebbero perdere il "filo" e renderebbero arduo un suo studio da parte dei nostri lettori. Ritengo, tuttavia, che la tua preparazione in materia sia notevole e che gli argomenti trattati nell'articolo possano esser "spezzettati" ed offerti in modo molto più semplice e (soprattutto) efficace agli utenti della nostra rivista. Telefonami, quindi, per concordare la nuova stesura dell'articolo. Tieni presente, inoltre, (e questo valga per gli altri lettori) che se c'è qualcosa che mi dà fastidio sono i dischetti che partono in auto-boot con modifica automatica dei colori dello schermo e del cursore...

TROPPO SEMPLICE
(Andrea Sepulcri - Udine)

Il programma, per C/128, che consente di gestire i principali comandi del drive, è un po' troppo semplice per esser proposto sulle nostre pagine. Il listato è tuttavia abbastanza "elegante" e ben fatto e denota una certa cura nella sua impostazione. Ti consiglio di approfondire i vari argomenti e, soprattutto, di sfruttare al massimo le potenti istruzioni che il C/128 offre ai suoi utilizzatori.

GIA' PUBBLICATO
(Claudio Cobianco - Cernusco)

Il listato inviato, che consente di disegnare funzioni matematiche sfruttando il nostro emulatore Gw-Basic, è molto simile, come risultati, ad altri programmi già pubblicati in precedenza.

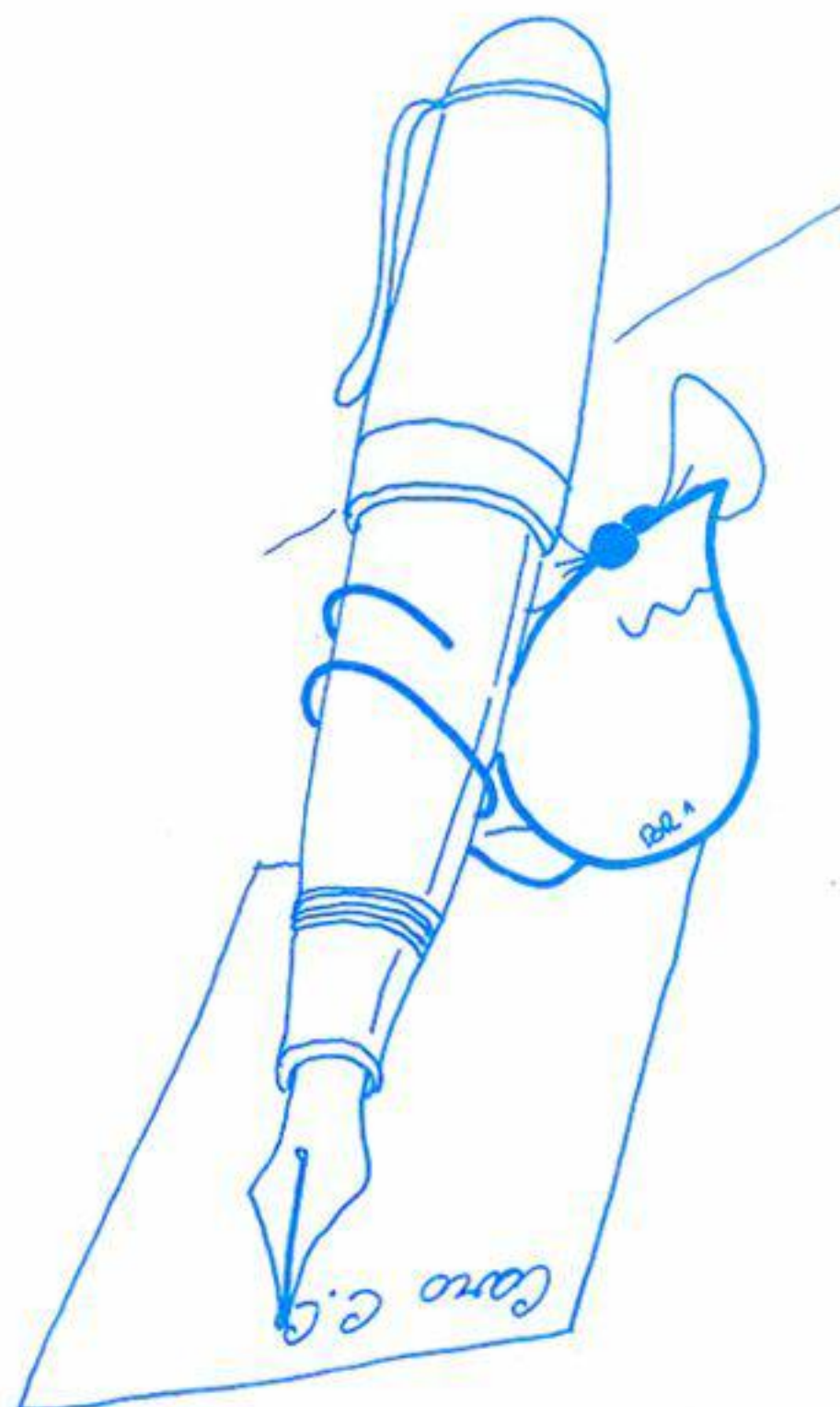
re bassissime quantità di energia elettrica; di conseguenza le batterie incorporate, benchè di modesta potenza, sono in grado di tenere in vita il programma per intere settimane.

Un C/64, e tutti i computer che necessitano di alimentazione di rete, consuma una quantità di energia relativamente elevata che scaricherebbe, in breve tempo, anche la batteria di un'automobile.

Per fare in modo che un computer sia sempre attivo (anche nei casi malaugurati di interruzione di energia elettrica) è necessario usare i cosiddetti gruppi di continuità.

Questi sono particolari apparati che, di norma, alimentano il sistema mediante la "solita" corrente della linea ENEL. Contemporaneamente, però, tengono sotto tensione una batteria che, in caso di interruzione di energia, entra automaticamente (ed immediatamente) in funzione garantendo il normale funzionamento dell'apparato.

Al termine dell'interruzione, l'energia viene di nuovo prelevata dalla linea, mentre la batteria viene ricaricata.

Si tratta, comunque, di situazioni di emergenza e di apparecchiature che, per un C/64 completo (computer, drive, monitor), possono superare le 500 mila lire.

Amiga 2000

Amiga 2000 è il collaboratore ideale per preparare facilmente ed in pochissimo tempo al video testi e grafici e riportarli poi su carta, su lucidi o diapositive. È uno strumento creativo semplice e immediato per realizzare soggetti fantastici in due o tre dimensioni con 4.096 colori. Qualunque sia la tua professione, con Amiga 2000 hai a disposizione le meraviglie dell'ambiente Amiga DOS e del sistema MS-DOS, con i quali potrai elaborare i tuoi progetti con una grafica ineguagliabile. In più, Amiga 2000 è Commodore: un marchio leader nel mondo degli home computer e dei sistemi professionali. Non a caso il primo PC (P.E.T.) è stato prodotto da Commodore, che può così vantare la più lunga esperienza nel settore.

- MS-DOS e XENIX sono marchi registrati della Microsoft Inc.
- UNIX è un marchio registrato della AT&T.
- Commodore e Amiga sono marchi registrati della Commodore Inc.

Amiga 2000.

Per te che vuoi un amico sulla scrivania.

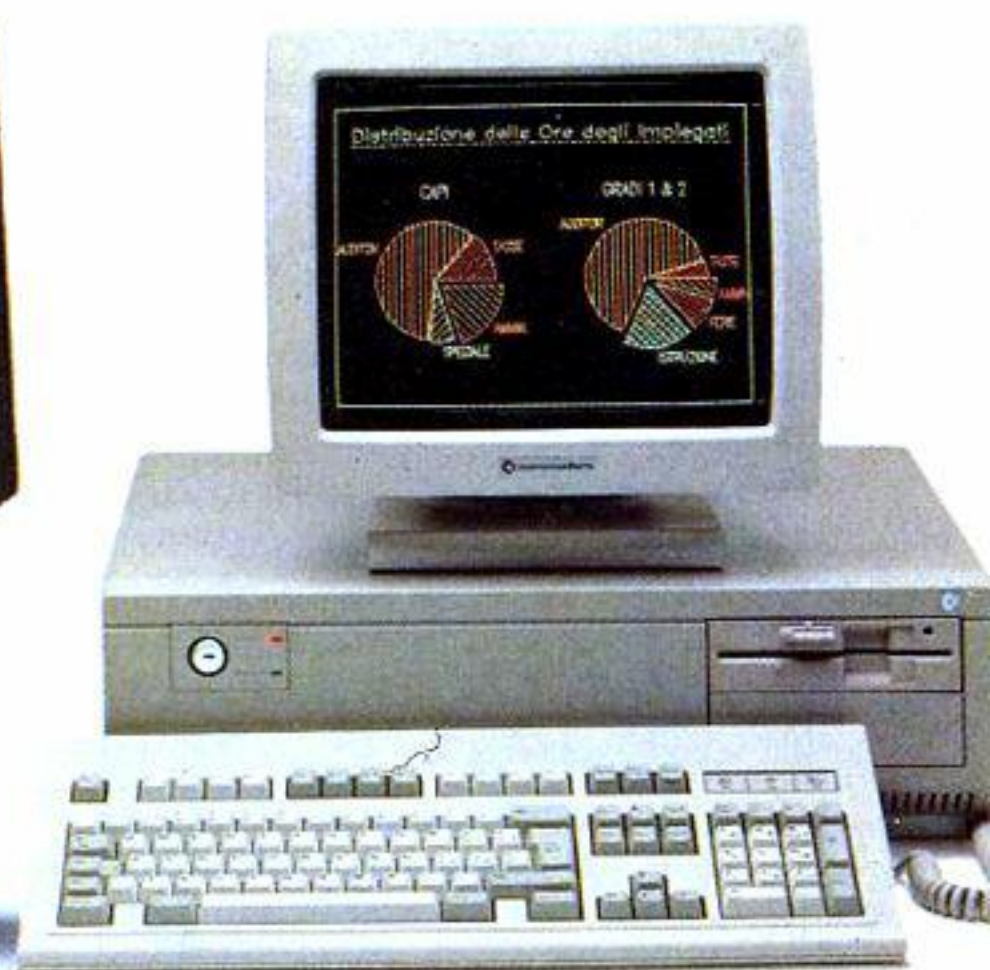
Amiga 2000.

Per te che vuoi lavorare con un creativo alla tua altezza.

Amiga 200(

Per te e per i tuoi progetti che non hanno bisogno solo del sistema MS-DOS.

C'è un Commodore p

Concessionario
Commodore
Sistemi Professionali

Lombardia

Milano
- A'ION - Via Bigli, 11
- AL RISPARMIO - V.le Monza, 204
- B.C.S. - Via Montegani, 11
- BRAHA ALBERTO - Via Pier Capponi, 5
- EDI CARED - Via Pietro Calvi, 20
- E.D.S. - C.so Porta Ticinese, 4
- FLOPPERIA - V.le Montenero, 31
- GI-SETTANTA - Via Burlamacchi, 4
- GIGLIONI - V.le Luigi Sturzo, 45
- LOGITEK - Via Golgi, 60
- MARCUCCI - Via F.lli Bronzetti, 37
- NEWEL - Via Mac Mahon, 75
- SUPERGAMES - Via Vitruvio, 38
- 68000 E DINTORNI - Via Washington, 91

Provincia di Milano
- ALL COMPUTER - Residenza Sassi, 312 - Milano 3 - Basiglio
- PENATI - Via Verdi, 28/30 - Corbetta
- S.A.T. - Via Milano, 24 - Nerviano
- IL CURSORE - Via Campo dei Fiori, 35 - Novate Milanese

Bergamo
- DERCOM - Via Borgo Palazzo, 65/A

Provincia di Bergamo
- COMPUTER SHOP - Via V. Veneto, 9 - Capriate San Gervasio

Brescia
- MASTER INFORMATICA - Via F.lli Ugoni, 10/B

Provincia di Brescia
- CAVALLI PIETRO - Via X Giornate, 14/B - Castrezzato
- DATA SYSTEM NEW - Via Gramsci, 33 - Concesio
- MEGABYTE - P.zza Maluezzi, 14 - Desenzano del Garda

Provincia di Como
- CIMA ELETTRONICA - Via Leonardo da Vinci, 7 - Lecco
- FUMAGALLI - Via Cairoli, 48 - Lecco
- RIGHI ELETTRONICA - Via Leopardi, 26 - Olgiate Comasco

Cremona
- MONDO COMPUTER - Via Giuseppina, 11/B
- PRISMA - Via Buoso da Dovara, 8

Provincia di Cremona
- EUROELETTRONICA - Via XX Settembre, 92/A - Crema

Provincia di Mantova
- CLICK - ON COMPUTER - S.S. Goietese, 168 - Goito

Pavia
- POLIWARE - C.so Carlo Alberto, 76

Provincia di Pavia
- LOGICA MAINT - Via Montegrappa, 32 - Vigevano

Provincia di Sondrio
- FOTONOVA - Via Valeriana, 1 - San Pietro di Berbenno

Varese
- IL CENTRO ELETTRONICO - Via Morazzone, 2
- SUPERGAMES - Via Carrobbio, 13

Provincia di Varese
- CURIO TRE - Via Ronchetti, 71 - Cavaria
- J.A.C. NUOVE TECNOLOGIE - C.so Matteotti, 38 - Sesto Calende

Piemonte

Cuneo
- ROSSI COMPUTERS - C.so Nizza, 42

Provincia di Cuneo
- PUNTO BIT - C.so Langhe, 26/C - Alba

Novara
- SOFTEAM - Via Locchi, 6

Provincia di Novara
- L.A.E. SOFTWARE - C.so Cavour, 46/59 - Arona
- ALL COMPUTER - C.so Garibaldi, 106 - Borgomane- ro
- ELLIOTT COMPUTER SHOP - Via Don Minzoni, 32 - Intra

Torino
- ALEX COMPUTERS - C.so Francia, 233/4
- DE BUG COMPUTER - C.so Vittorio Emanuele II, 22
- DESME UNIVERSAL - Via San Secondo, 95
- IL COMPUTER - Via Nicola Fabrizi, 126
- STAF - C.so Regina Margherita, 97

Provincia di Torino
- DIAM INFORMATICA - C.so Francia, 146 BIS - Cascine Vica - Rivoli

Provincia di Vercelli
- C.S.I. TEOREMA - Via Losana, 9 - Biella
- CHIP - Via Martiri della Libertà, 47 - Cossato

Veneto

Belluno
- UP TO DATE - Via Vittorio Veneto, 43

Padova
- SARTO COMPUTER - Via Armistizio, 79

Trentino Alto Adige

Bolzano
- COMPUTER POINT - Via Roma, 82

Provincia di Bolzano
- ELEKTRO TAPPEINER - P.zza Principale 90 - Silandro

Trento
- CRONST - Via G. Galilei, 25

Friuli Venezia Giulia

Pordenone
- SIEL - Via Colonna, 45

Udine
- CO.R.EL. ITALIANA - Via Tavagnacco, 91
- MOFERT 2 - Via Leopardi, 21

Liguria

Genova
- ODEL - Via Orsini, 4 R
- SALS INFORMATICA - Via G. D'Annunzio, 2

Emilia

Bologna
- MINNELLA ALTA FEDELTÀ - Via Mazzini, 146/2
- SINOPIA INFORMATICA - V.le Pietramellara, 11

Provincia di Bologna
- S.C. COMPUTERS - Via Enrico Fermi, 4 - Castel San Pietro
- S.P.E. INFORMATICA - Via di Mezzo Ponente, 385 - Crevalcore

Modena
- VIDEO VAL WILLY COMPUTERS - Via Canaletto, 223

Provincia di Modena
- NEW MEDIA SYSTEM - Via Roma, 281 - Soliera

Parma
- NEW LIST COMPUTER - Via Nazario Sauro, 9

Reggio Emilia
- COMPUTERLINE - Via San Rocco, 10/C
- POOL SHOP - Via Emilia S. Stefano, 9/C

Romagna

Repubblica di San Marino
- A.C.S. - Via Nonagualdaria - Cailungo

Ferrara
- BUSINESS POINT - Via Carlo Mayer, 85

Provincia di Forlì
- TOP BIT - Via Veneto, 12 - Forlimpopoli
- EASY COMPUTER - Via Lagomaggio, 50 - Rimini
- NUMERO - Via Battaglini, 21 - Rimini

Provincia di Ravenna
- E.T.S. - Via Saffi, 1 - Alfonsine
- P.L.Z. INFORMATICA - P.zza Sercognani, 6 - Faenza

Toscana

Arezzo
- DELTA SYSTEM - Via Piave, 13

Firenze
- M.T.S. DISTRIBUZIONE - Via di Novoli, 64 A
- TELEINFORMATICA TOSCANA - Via Bronzino, 36

Grosseto
- COMPUTER SERVICE - Via Dell'Unione, 2

Livorno
- ETA BETA COMPUTER - Via S. Francesco, 30
- FUTURA 2 - Via Cambini, 19

Provincia di Lucca
- IL COMPUTER - V.le Colombo, 216 - Lido di Camaiore

Pisa
- ELECTRONIC SERVICE - Via della Vecchia Tranvia, 10

Pistoia
- ELECTRONIC SHOP - Via degli Scalzi, 3

Siena
- RENATO BROGI - P.zza Gramsci, 28

Provincia di Siena
- ELETTRONICA di BIFOLCHI - Via di Gracciano nel Corso, 111 - Montepulciano
- ELETTROMERCATO di BURRINI - Via Toscana, 6 - Monteriggioni

Umbria

Perugia
- MIGLIORATI - Via S. Ercolano, 3
- STUDIO SYSTEM - Via R. D'Andreotto, 49/51

Lazio

Roma
- D.R.R. - Via Giovanni Giorgi, 6

Abruzzo

Teramo
- MAX CENTER - V.le Crispi, 68

Puglia

Bari
- ARTEL - Via Guido d'Orso, 9
- COMPUTER'S ARTS - V.le Meucci, 12/B
- ELETTRONICA SISTEMI - V.le della Repubblica, 67/69
- PAULICELLI SABINO & FIGLI - Via Fanelli, 231/C

Campania

Napoli
- DARVIN - Calata San Marco, 25
- DPC INFORMATICA - Via E. Nicolardi, 129
- GENERAL COMPUTERS - Via Bernini, 101
- SPY - Via Domenico Fontana, 135
- TOP VIDEO - TOP COMPUTER - Via S. Anna dei Lombardi, 12

Provincia di Napoli
- ITALIANA SOFTWARE - Via Zara, 11 - Acerra
- NUOVA INFORMATICA SHOP - Via Libertà, 185 - Portici

Provincia di Salerno
- COMPUTER WORLD - Via Quarto, 6 - San Marzano sul Sarno

Calabria

Cosenza
- SIRANGELO COMPUTER - Via N. Parisio, 25

Reggio Calabria
- CONTROL SYSTEM - Via S. Francesco da Paola, 49/D-E
- SYSTEM HOUSE - Via Fiume ang. Palestrino, 1

Provincia di Reggio Calabria
- COMPUTER SHOP - Via Matteotti, 48/50 - Locri

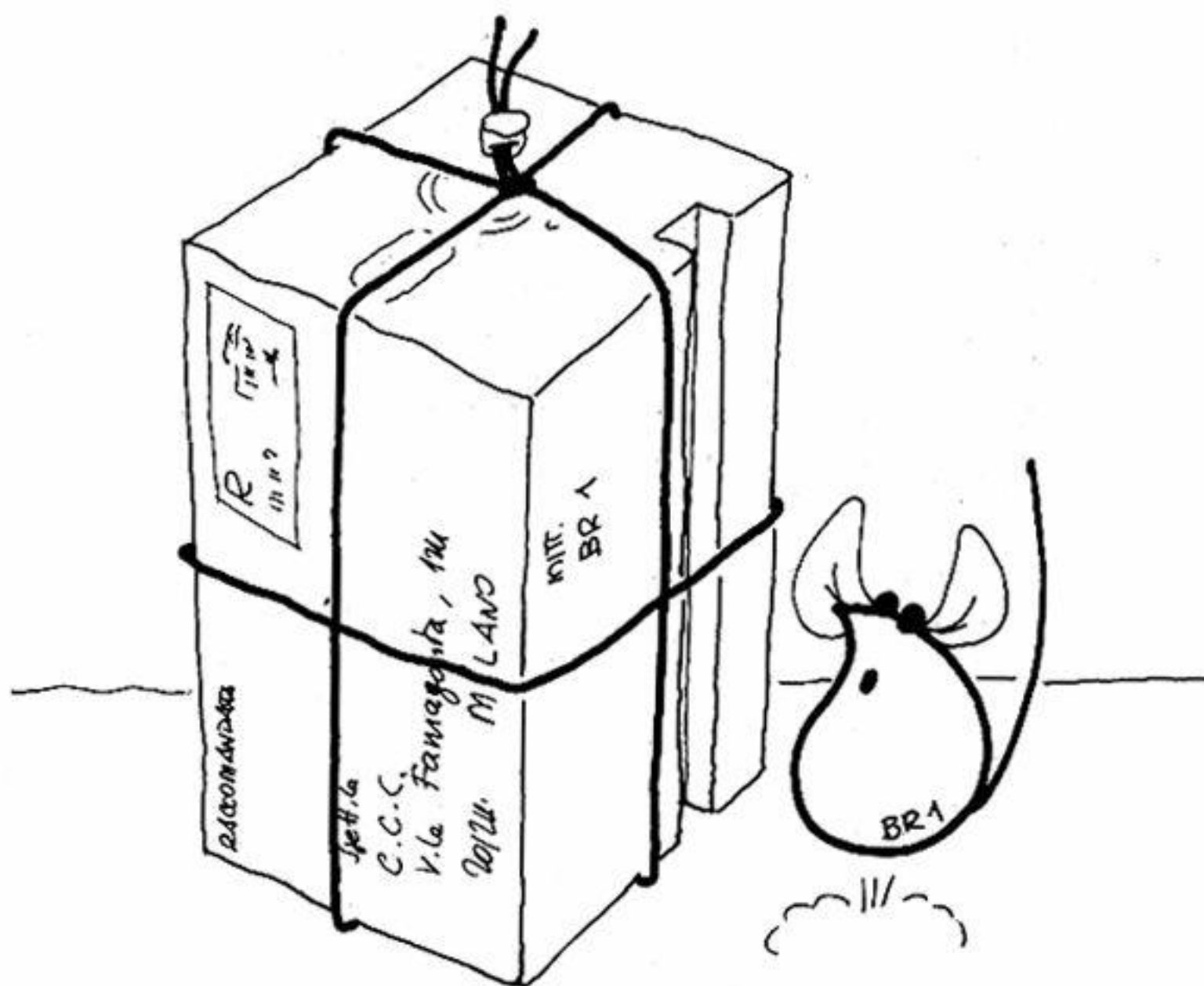
Sicilia

Provincia di Enna
- ITALSOFT - Via Dottor Palazzolo - Agira

Palermo
- C.H.P. - Via Libertà, 95

cambia in Commodore®

Esigi sempre la garanzia della Commodore Italiana S.p.A.

Elenco aggiornato al 3 Aprile 1989

INTUITION

☐ **Sto per abbandonare il C/64 per passare ad Amiga e vorrei sapere che cosa è Intuition.**
(Alberto Gili - Torino)

• Con lo "strano" termine segnalato dal nostro lettore viene indicata l'interfaccia utente utilizzata dal sistema Amiga.

In altre (e più semplici) parole, Intuition è quel complesso di routine (rigorosamente in linguaggio macchina) che permettono un uso estremamente facilitato del potente computer.

Ad Intuition, ad esempio, appartengono le routine che consentono di aprire una finestra cliccando due volte sulla corrispondente icona, di effettuare copie di file "catturando" un'icona e trasportandola, con il mouse, in un'altra finestra e così via.

Intuition, insomma, è quell'insieme di procedure (= interfaccia) che consente all'utente di "intuire" la via giusta da seguire per ottenere determinati risultati, limitandosi, per di più, a premere semplicemente un tasto del mouse.

CONCORSI

☐ **Perchè, di tanto in tanto, non bandite concorsi per i lettori?**
(Roberto Gallo)

• Purtroppo vi sono due impedimenti; il primo è di tipo burocratico dal momento che la procedura per avere i vari permessi è lunga (e noiosa).

In secondo luogo siamo (parzialmente) contrari ad effettuare sorteggi di tagliandi da inviare in Redazione (che c'entra la fortuna con l'informatica?). Ci è capitato, nel passato, di spronare i nostri lettori a produrre software di un certo pregio, ma l'offerta di materiale è risultata praticamente identica a quella che di solito viene proposta.

In futuro, però, chissà...

PLOTTER

☐ **Ho alcuni problemi con il plotter Commodore 1520 perchè vorrei usarlo con l'ottimo package "Totocalcio" da Voi edito. Che modifiche devo effettuare?**
(Guido De Silvestro - Trieste)

• La modifica è relativamente semplice dal momento che il numero che contraddistingue la periferica è 6.

E' infatti risaputo che ogni periferica Commodore è individuata da un numero: 8 è il drive, 4 la stampante, 3 il monitor, 1 il registratore e 2 l'interfaccia Rs-232.

Per fare in modo che l'utente dei piccoli Commodore potesse utilizzare, contemporaneamente, sia il plotter sia la stampante, i progettisti decisero di assegnare, al plotter, il numero di device 6 proprio per non creare interferenze con la stampante (4).

Pertanto è necessario rintracciare, nei programmi Basic che operano con una stampante, i comandi relativi all'apertura del colloquio con tale periferica...

```
Open 4, 4
```

...e sostituirlo con...

```
Open 4, 6
```

I successivi comandi di Print# non vanno modificati.

Non tutto, però, potrebbe filar liscio: alcuni comandi specifici per le stampanti 803 - compatibili, infatti, non hanno significato per il plotter, e viceversa.

In ogni caso tieni conto che non è possibile danneggiare gli apparecchi anche nel caso di errata impostazione di comandi.

PER UNA POTENZA IN PIU'

Ho notato uno strano fenomeno che vorrei sottoporre alla vostra attenzione...
(Livio Ariboli - Arezzo)

• "...digitando il semplice listato, prosegue Livio...

```
10 h = 27: k = 2
20 r = h / (3 ↑ k)
30 print r
40 print int(r)
```

...si ottiene, come risposta, 3 (riga 30) e 2 (riga 40). Il valore 3 non dovrebbe di per sè essere già intero?" si chiede costernato il nostro lettore.

La spiegazione dello strano fenomeno è relativamente semplice e riguarda (ancora una volta) l'approssimativa gestione dei numeri in virgola mobile (in generale) e delle potenze (in particolare).

L'algoritmo presente nel C/64, infatti, prevede due pesi e due misure. Quando viene visualizzato il "3", alla variabile R è associato un valore, in virgola mobile, che non è quello che verrebbe assegnato digitando, direttamente, R=3.

I cinque byte incaricati di memorizzare la variabile, infatti, contengono un valore

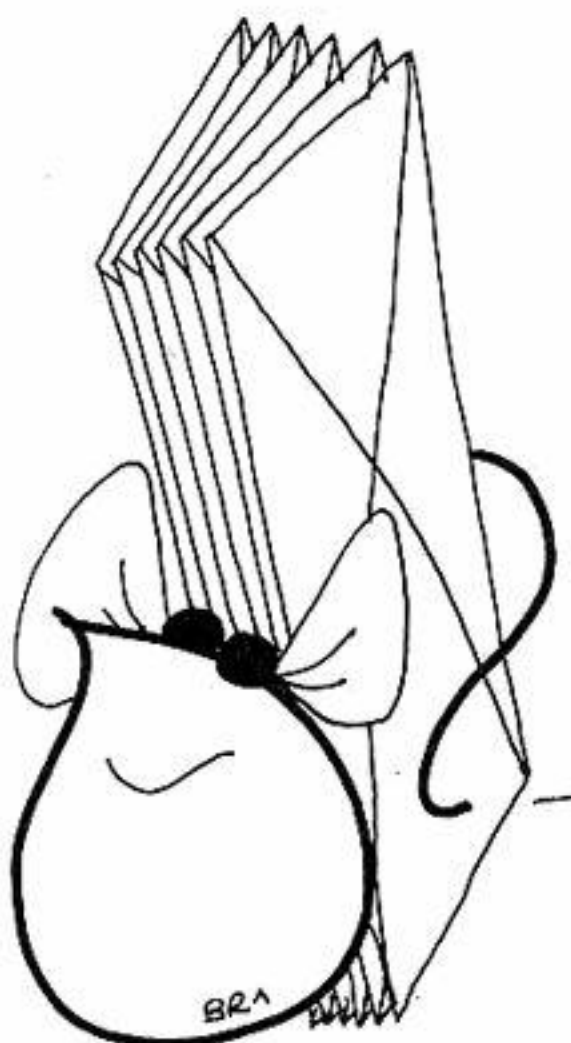
decimale "molto" vicino a 3 che solo nella fase di visualizzazione appare come intero: in realtà è memorizzato come decimale, inferiore di pochissimo a 3.

Elaborando successivamente tale valore (ed è il caso della riga 40 che tenta di "estrarre" il valore intero di R) i cinque byte considerano il numero per quello che è, vale a dire decimale e "minore" di 3.

Per evitare simili inconvenienti (che compaiono soprattutto con potenze dispari di numeri dispari) è spesso sufficiente far compiere un'operazione "inversa" alla precedente.

Esempio:

```
10 h = 27: k = 2
20 r = h / (3 ↑ k)
30 r = r ↑ 1
40 print r: print int(r)
```

La riga 30, in questo caso, eleva alla potenza unitaria la variabile R che rimane identica a se stessa. La provvidenziale operazione, ingannevolmente inutile, converte la variabile R da "apparentemente" intera a "intera a tutti gli effetti"; e la riga 40 lo dimostra.

CP/M E DRIVE 1581

☐ Su un precedente numero di C.C.C. avete affermato che non è possibile formattare, con C/128 e drive 1581, un dischetto da 3.5 pollici in modalità CP/M. Io ci sono riuscito...

(F. Marozzo - Presezzo)

• La procedura da seguire (prosegue il nostro lettore) è semplicissima:

```
Open 15, DV, 15
Print #15, "U0" + Chr$( 134) + Chr$(2)
+ Chr$(40)
Close 15
```

Ringraziamo F. Marozzo per la precisazione e ne approfittiamo per avvertire gli interessati che un nostro collaboratore, su altre indicazioni suggerite dallo stesso lettore, sta sperimentando altre procedure per utilizzare il drive 1581 in modo CP/M. Non appena ne sapremo di più, ne riparleremo.

risposte rapide

INPUT SONORO PER C/16
(Pasquale Cammarano - S. Germano)

Non c'è in giro documentazione sufficiente per elaborare, con un C/16, un segnale audio proveniente dall'esterno. Se qualche lettore ha sviluppato un programma specifico (purchè non molto lungo) possiamo esaminarlo per un'eventuale pubblicazione.

COPIARE
(Cristiano Lanzano - Torino)

Tra gli appassionati (e presso i negozi specializzati) circolano numerosissime routine che consentono di effettuare copie di interi dischi o di singoli file. Prova a contattare gli inserzionisti il cui nome compare sulle nostre pagine.

SENZA MANUALE
(Luca Lostorto - Terni)
(Andrea)

Non vendiamo manuali Commodore nè cartucce o accessori descritti nelle pagine pubblicitarie. Rivolgi, piuttosto, le tue lamentele presso il rivenditore che ti ha venduto l'adattatore telematico privo di manuale!

TUTTO A POSTO
(Emilio Soresi - Partinico)

La lucetta verde del drive DEVE accendersi durante le operazioni di registrazione e lettura dei file per spegnersi ad operazione conclusa; il tuo drive, quindi, funziona perfettamente (sarebbe guasto in caso contrario!).

FORMULA 3-D
(Marco Alzetta - Pordenone)

Le relazioni matematiche per disegnare in prospettiva in hi-res sono piuttosto complesse. Una spiegazione (ma solo per espertissimi) è riportata nel disassemblato commentato presente sul fascicolo "Commodore speciale: routine grafiche" che, appunto, è interamente dedicato all'implementazione di un gruppo di routine l.m. per disegnare, sul C/64, in prospettiva. Il fascicolo è disponibile presso il nostro servizio arretrati (tel. 02/84.67.348)

INTERFACCIA AMIGA 1541
(da alcune telefonate)

L'interfaccia descritta sul N.60 sembra non funzionare correttamente con alcuni esemplari di Amiga e/o drive 1541. Ne riparleremo non appena scoperta la causa dell'inconveniente.

POCHE RICHIESTE
(Luciano Vendramin - Chivasso)

Purtroppo l'espansione di memoria 1764 per C/64 non ha avuto successo e la richiesta di trattare argomenti ad essa inerenti è tanto modesta da scoraggiare la pubblicazione di programmi specifici.

POCO CHIARO
(Giovanni Katurano - Benevento)

La lettera inviata è poco chiara e, in certi casi, addirittura illeggibile. Ti prego di scrivere le domande in modo più comprensibile.

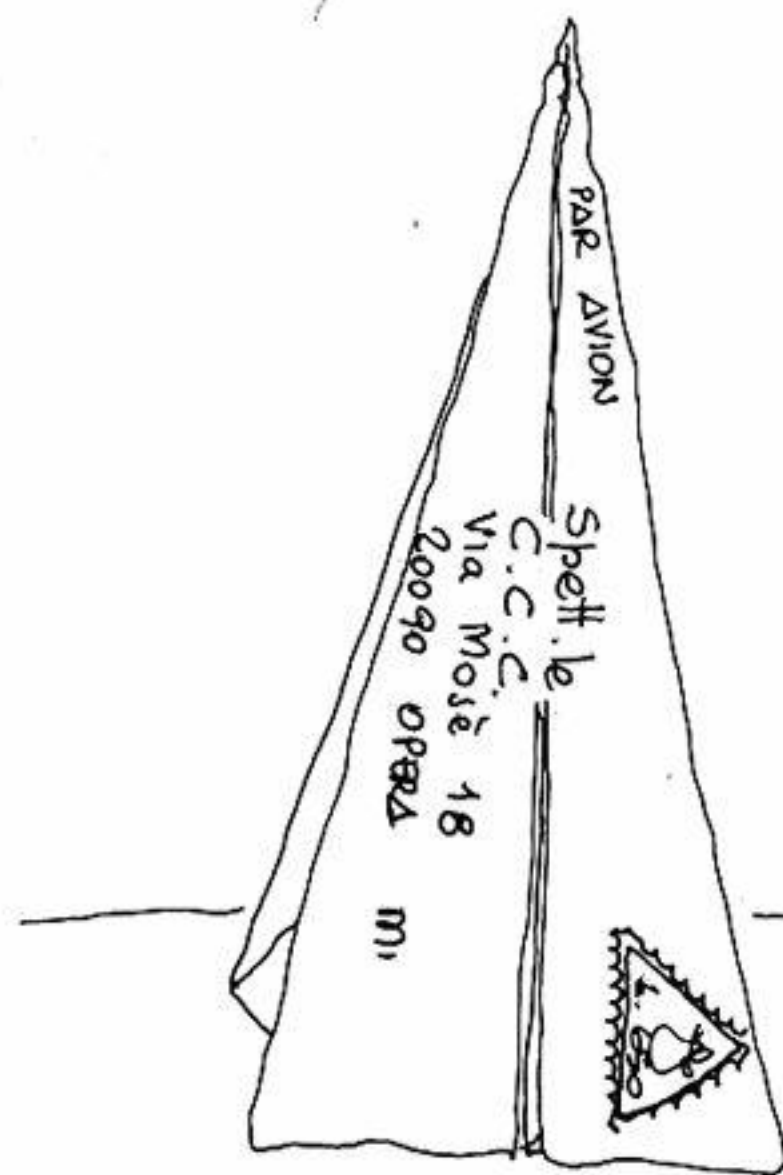
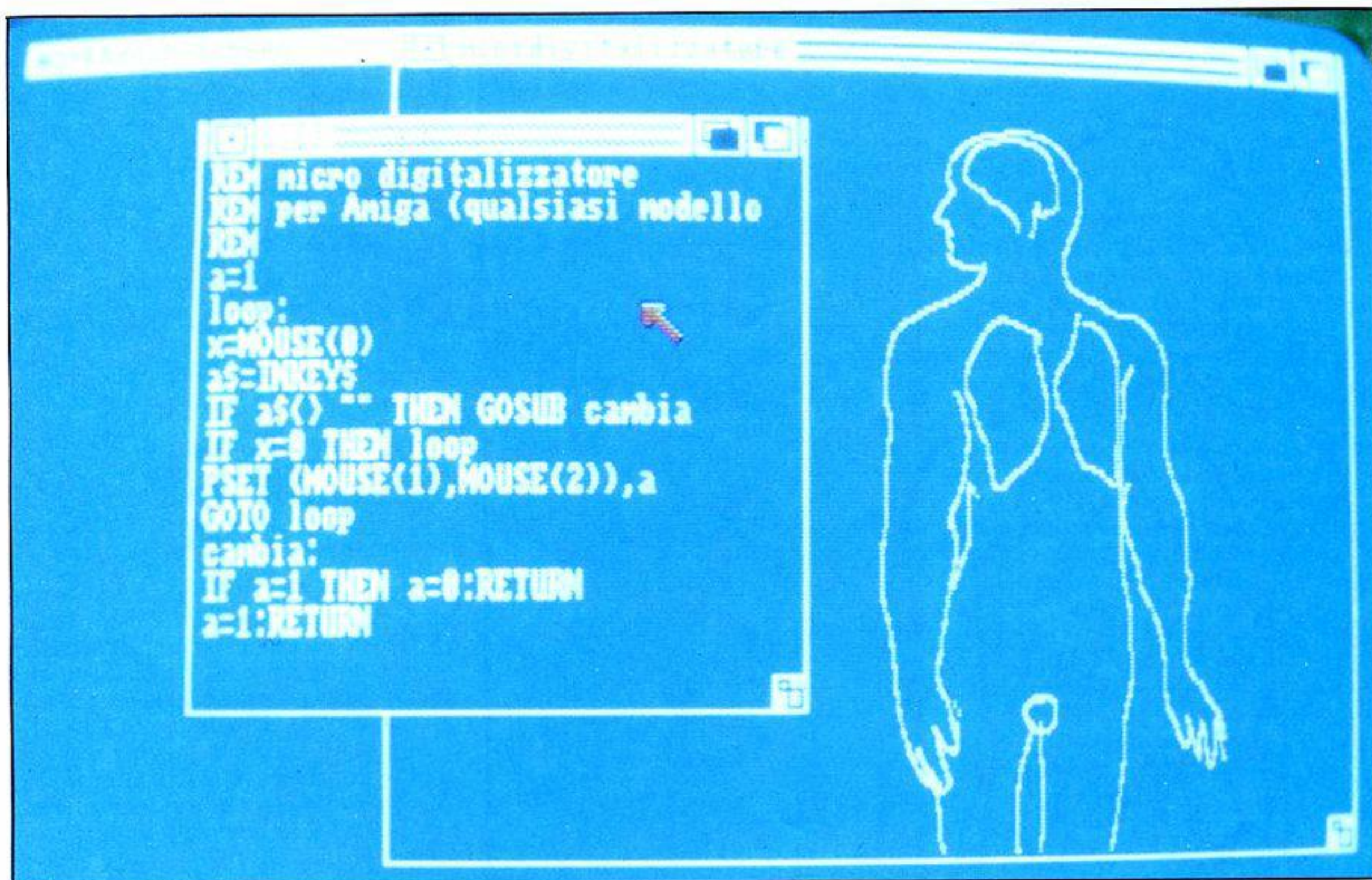
NON SU CARTA
(Antonio Medagliani - Stradella)
(Mario Saggese - Benevento)
(Giovanni Giovannelli - Rieti)

Non possiamo prendere in considerazione programmi inviati solo su carta. E' indispensabile inviarli su disco (si escludono in modo assoluto le cassette).

SENZA PROBLEMI
(Alessio Isaia - Trieste)

Se non hai problemi economici penso che sia meglio acquistare un Amiga anzichè dotare il tuo C/128 di drive.

COINCIDENZE
(Marco Restuccia - S.M.C. Vetere)

Se il tuo C/64 si è improvvisamente guastato durante il funzionamento di un videogame si tratta di una semplice coincidenza. Non è possibile danneggiare un computer via software.

UNA SFIDA GRAFICA

Una graffetta, un po' di nastro adesivo e qualche istruzione Basic: ecco come realizzare un rilevatore di dati per Amiga

di **Alessandro de Simone**

Osservando il funzionamento di una macchina a controllo numerico non è difficile stupirsi per le straordinarie potenzialità offerte dalla moderna tecnolgia.

In pratica si tratta di seguire i contorni della sagoma che si vuol realizzare ed impartire, alla macchina utensile, i comandi idonei per la fabbricazione vera e propria del pezzo.

Il sistema, in toria molto semplice, si basa sull'utilizzo di uno speciale tecnigrafo sul cui tavolo viene fissato il disegno della sagoma, in scala 1:1.

La testa del tecnigrafo non possiede penne o righe, ma è dotata di una lente di ingrandimento in grado di evidenziare con maggior precisione i punti più importanti della sagoma da riprodurre.

Inutile dire che i binari su cui scorrono i bracci dello speciale tecnigrafo sono dotati di sensori elettrici molto sofisticati, in grado di individuare spostamenti minimi della testa (dell'ordine di decimi di millimetro) e di comunicarli, ovviamente, al computer con cui sono collegati.

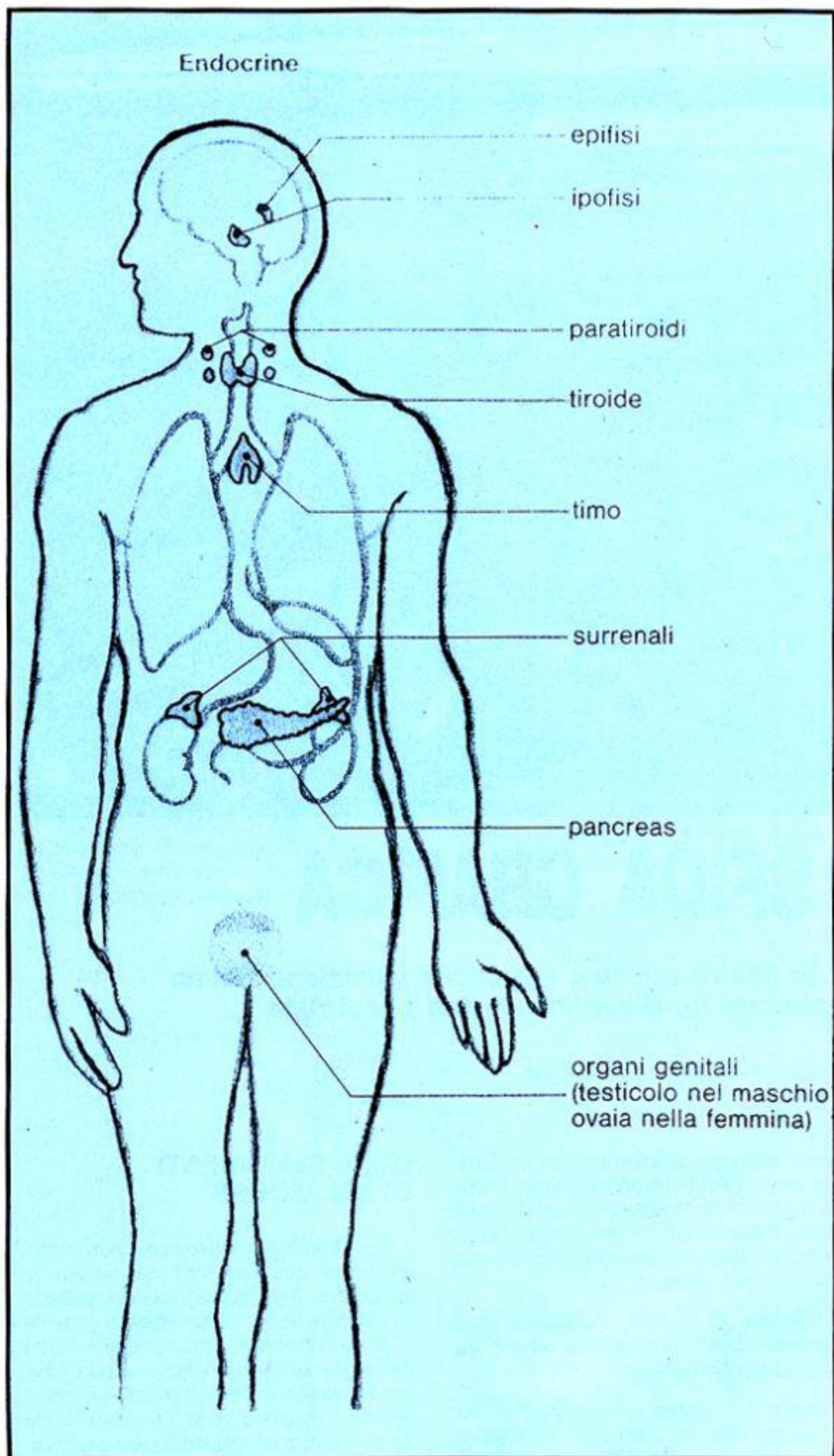
I sensori di cui stiamo parlando sono piuttosto costosi e di difficile reperibilità, almeno per un hobbista.

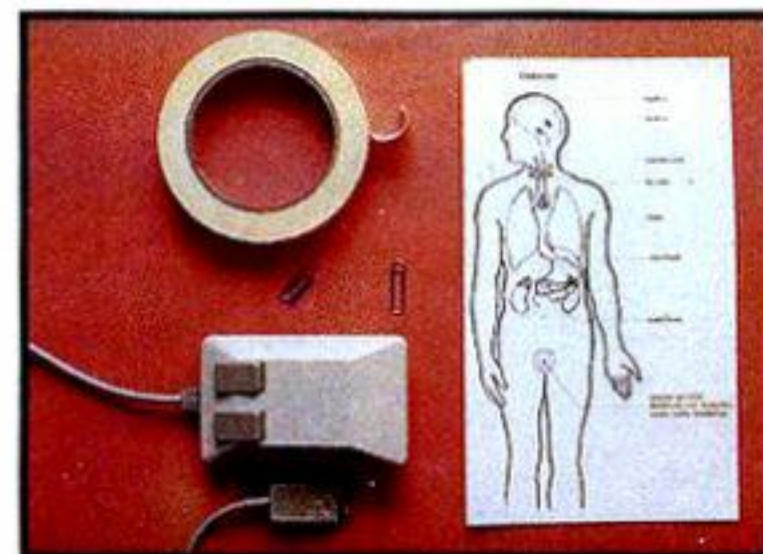
Inoltre è necessario un programma specifico che elabori i dati inviati dal tecnigrafo e li "traduca" nelle corrispondenti coordinate da elaborare ulteriormente.

Un sistema di rilevamento dati sembra quindi essere roba da professionisti e, in ogni caso, un'attrezzatura di notevole costo... Ma ne siamo proprio sicuri?

QUEL FENOMENO DI UN MOUSE

Con Amiga (ma il discorso si può trasferire ad un qualsiasi altro sistema basato sull'utilizzo di un mouse) abbiamo provato a realizzare un rozzo, ma efficace, sistema di acquisizione dati. Immaginate la nostra meraviglia quando ci siamo accorti che, pur se con qualche limitazione, il sistema funziona, eccome, perfino a mano libera! La procedura da seguire è talmente semplice che chiunque può realizzarla con il proprio computer, senza alcun pericolo per il mouse.

Si tratta, infatti, di procurarsi semplicemente un po' di nastro adesivo ed una graffetta (del tipo che si usa negli uffici per

tenere assieme più fogli di carta); è possibile, invece della graffetta, usare un fiammifero o un qualsiasi mini-oggetto, purchè molto sottile, possibilmente curvabile e lungo circa 6 centimetri.

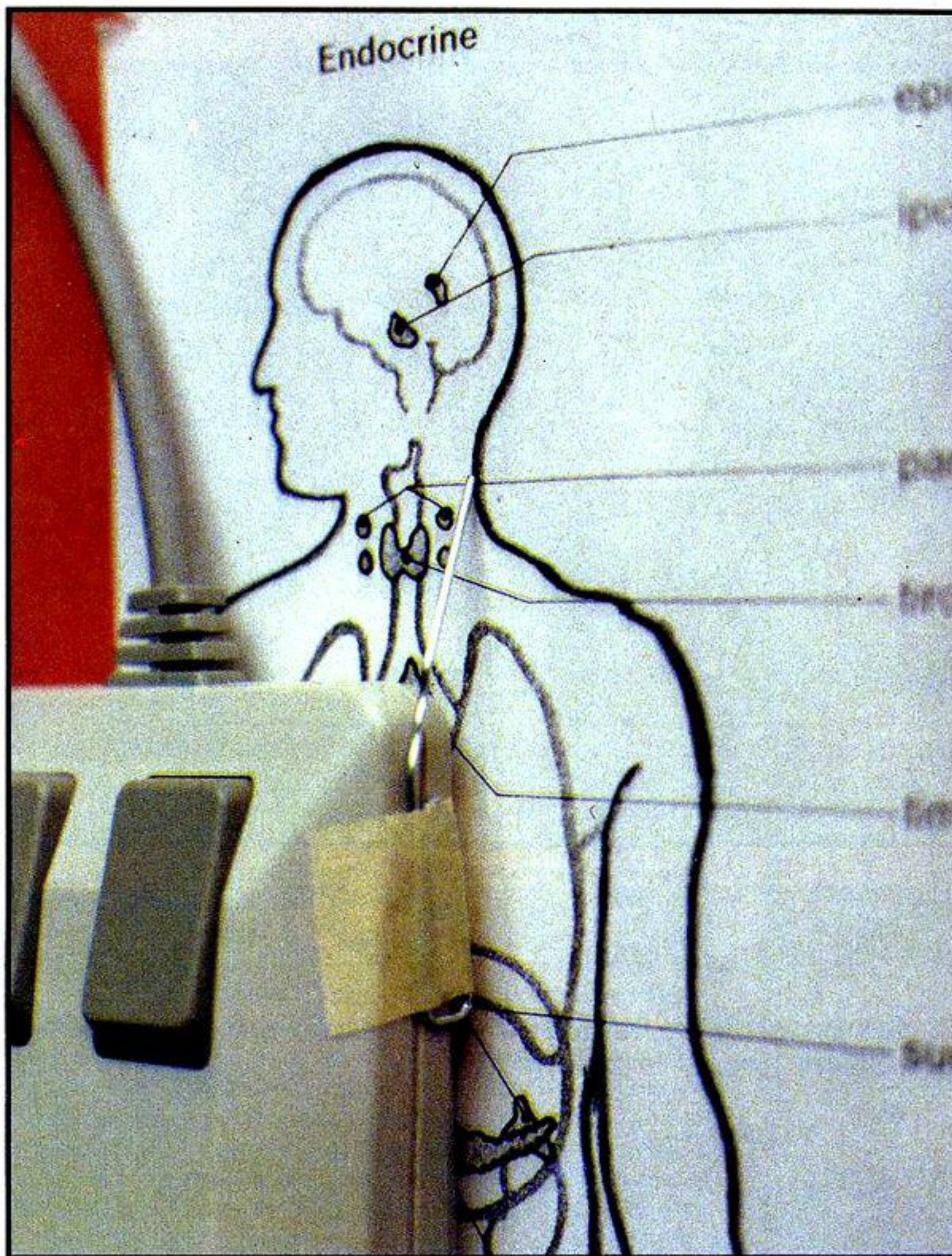
La graffetta, opportunamente "aperta" e sagomata, verrà posta sul lato superiore del mouse (a destra o a sinistra, non importa) attaccandola con il nastro adesivo in modo che la punta della graffetta possa seguire il profilo del disegno da riprodurre su schermo.

Procuratevi ora il disegno da riprodurre su video (per le prove ci siamo serviti della fotocopia tratta da un libro scientifico) e fissatelo, con nastro adesivo, sul piano che di solito usate per manovrare il mouse.

Accendete Amiga, caricate Workbench, aprite la finestra "Prefs" e cliccate l'icona "Preferences". Comparsa la tipica finestra, modificate l'escursione del mouse (indicata con "velocità mouse" pagina 6-2 del manuale), portandolo dal valore 1 (di default) a 4. Tale accorgimento consentirà al mouse di operare su una superficie più grande, per lavorare su disegni di grande dimensione con conseguente maggior dettaglio.

Cliccate ora su "Use"; vi accorgerete, appunto, che per andare da una parte all'altra dello schermo è necessario far compiere un notevole tragitto al nostro mouse.

Cancellate pure le finestre di Workbench, inserite il disco "Extras", caricate il "Basic" e digitate il seguente, banalissimo programmino:

```
a=1
loop:
x = Mouse (0)
a$=inkey$
if a$ < [◇"" then gosub cambia
if x=0 then loop
Pset (mouse(1), mouse(2)), a
Goto loop
cambia:
if a=1 then a=0: Return
a=1: Return
```

Se disponete di uno dei tantissimi programmi grafici per Amiga (del tipo Deluxe-Paint) potete fare a meno di digitare il programmino che, tuttavia, descriviamo per la gioia dei principianti:

La variabile "a" funziona da deviatore. Quando vale 1 consentirà di disegnare; se vale zero cancellerà la zona del video sulla quale "passa" il mouse.

La variabile "a" cambia stato tutte le volte che si preme un tasto della tastiera; la subroutine "Cambia" è sufficientemente chiara e non necessita di spiegazioni.

Se il tasto sinistro del mouse non è premuto, è possibile spostare la freccia in lungo e in largo, senza "disturbare" il disegno. Se, invece, si tiene premuto il tasto sinistro del mouse, la freccia lascerà una scia di colore bianco (se "a" vale 1); altrimenti lascia una scia dello stesso colore del fondo (se "a" vale zero): in quest'ultimo caso, in altre parole, verrà cancellata la zona video sulla quale si posiziona il mouse premendo il tasto sinistro.

Provate il programma tracciando linee e poi cancellandole, in modo da impratichirvi prima di passare ad usare il mouse come rilevatore di dati.

DIGITALIZZIAMO

Il termine "digitalizzare", in effetti, è qui usato impropriamente, ma rende bene l'operazione che andremo a compiere.

Dunque, prendete il disegno che volete riprodurre su video e fissatelo, con quattro pezzetti di nastro adesivo, alla porzione di tavolo sulla quale opererà il mouse.

Fate partire il programma e posizionate la freccia del mouse all'incirca in alto al centro dello schermo.

Sollevando ora il mouse (per impedire alla freccia di mutar posizione sul video) riponetelo delicatamente sul foglio di carta in modo che la punta della graffetta (che in precednza avete sistemato con nastro adesivo) coincida con la parte superiore del disegno da riprodurre.

A questo punto, FACENDO BEN ATTENZIONE A NON RUOTARE IL POLSO DURANTE IL MOVIMENTO, seguite il profilo del disegno tenendo abbassato il pulsante sinistro del mouse: sullo schermo dovrebbe comparire lo stesso profilo.

E' bene sottolineare che il profilo riprodotto non può esser perfettamente identico all'originale, per due motivi.

Il primo è dovuto alla imprecisone del mouse stesso; il secondo è dovuto al fatto che, pur se si presta la massima attenzione, è praticamente impossibile evitare rotazioni del polso che, anche se minime, contribuiscono a "disassare" il disegno con conseguenze più o meno vistose.

Con un po' di pazienza, tuttavia, potrete verificare che seguire i contorni di un disegno è più semplice di quanto sia possibile immaginare. Se, poi, usate un vero Tool grafico, potrete apportare tutte le correzioni del caso.

SFIDA AI LETTORI

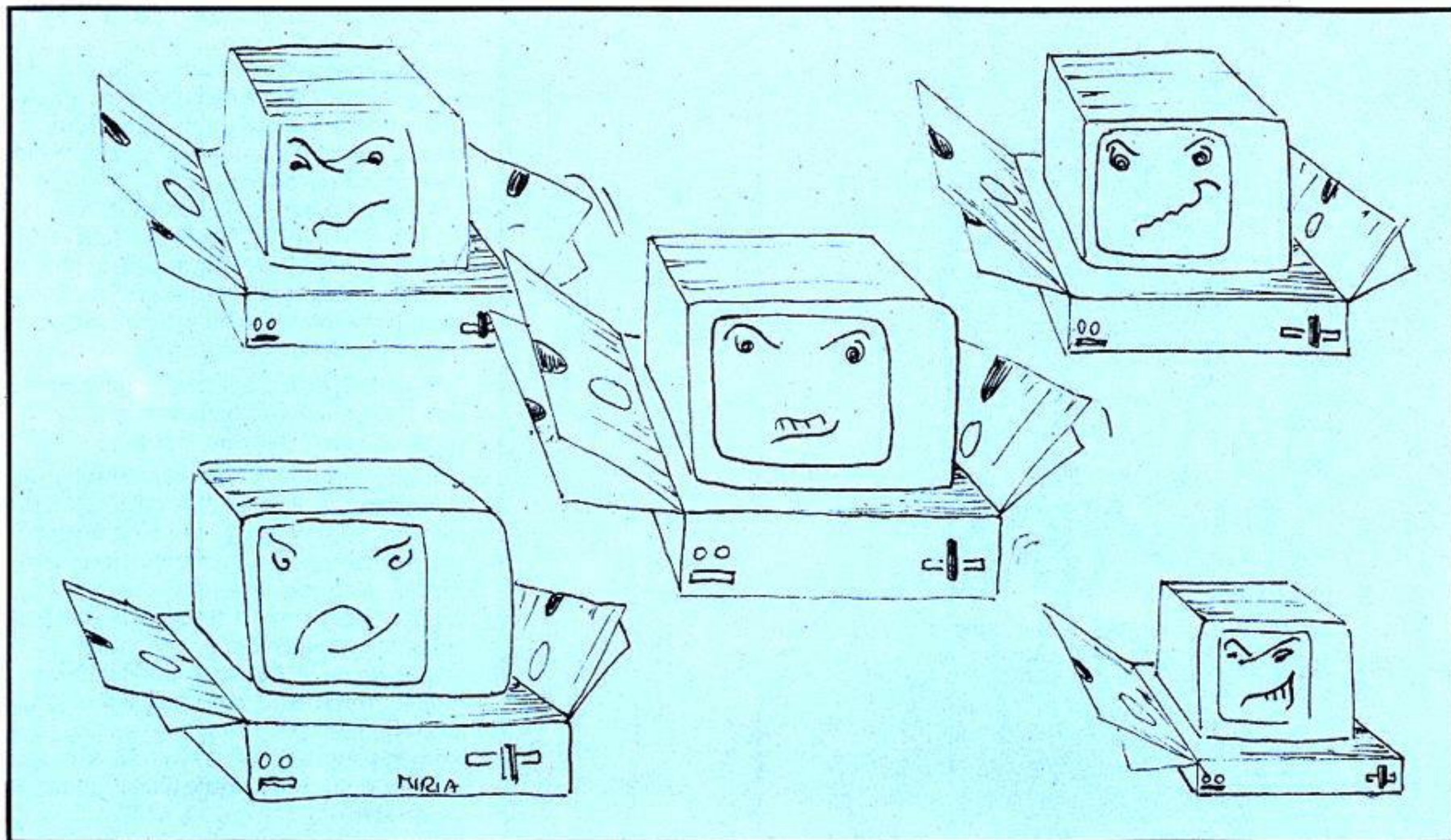
Il motivo principale dell'imprecisione è dovuto al fatto che è impossibile mantenere il mouse sempre parallelo a se stesso, soprattutto se il disegno richiede continui spostamenti in diagonale.

Con alcuni accorgimenti, però, è possibile modificare il percorso: con un occhio al disegno ed uno allo schermo potrete "barare" forzando lo spostamento della freccia (sollevando il mouse e riposizionandolo opportunamente, facendo scorrere, sotto, un secondo foglio di carta per muovere la sfera ma lasciando immobile il mouse, e così via).

Se, però, si sistema il mouse in modo più "rigido" il pericolo di scostamento dagli assi principali si riduce notevolmente.

Ciò si può realizzare, ad esempio, fissando il mouse sulla testa di un tecnigrafo o realizzando, appositamente, un "qualcosa" che lo costringa a non ruotare durante gli spostamenti.

L'idea di base l'abbiamo data. Tocca ora a voi realizzare un sistema valido ed inviarci i vostri progetti.

I migliori tra questi verranno citati sulla nostra rivista, a patto che i prototipi siano fotografati con la massima chiarezza (megio se in diapositiva) ed inviati all'indirizzo riportato nella pagina in cui compare l'indice di questo stesso fascicolo.

LA GUERRA DEI NUCLEI

Che cosa accade quando due programmi, residenti nello stesso computer, dichiarano aperte le ostilità?

di **Fabio Simeoni**

L'epidemia informatica scoppiata nel 1987 in California, nota con il nome di virus mangiasoftware, rappresenta, come spesso sulle stesse pagine di questa rivista è stato sottolineato, un argomento di sicuro interesse.

Si è parlato molto di come riconoscerlo, di come prevenirlo o di come limitarne i danni; si è analizzata efficacemente la sua logica di funzionamento, di duplicazione, e si è anche pensato di darne semplici esempi applicativi, con intenzioni esclusivamente didattiche (chi potrebbe dubitarlo?), in sintonia con la filosofia secondo la quale "fare" equivale ad "imparare", conoscere e quindi prevenire.

Raramente, però, si trovano accenni sulle origini di questa famiglia di programmi "virulenti", che ci introducono in una delle leggende più accattivanti della storia informatica.

STORIA

Un programmatore di una certa Società americana scrisse un programma, ovviamente in linguaggio assemblatore, che presentava la singolare caratteristica di autoduplicarsi nel momento in cui veniva mandato in esecuzione: non aveva (almeno all'inizio) altri scopi fuorchè quello citato.

Il programma, a cui fu imposto il nome di "Creeper" (il rampicante), ben presto si diffuse con la subdola rapidità specifica di un qualsiasi virus, su tutta la rete computerizzata di quella Società; riuscì ben presto ad infestarla senza più possibilità di controllo. Come mezzo difensivo fu scritto un altro programma, complementare del primo, che avesse l'unica funzione di cancellare tutti i Creeper che incontrava quando veniva lanciato.

Reaper, cioè "Falciatore", venne infatti chiamato il nuovo programma che assolse a dovere il suo compito e riportò la situazione alla normalità. La storia, evidentemente, attirò l'interesse di un giovane ricercatore dell'università dell'Ontario, abbastanza creativo e intuitivo per non accorgersi che, da quel piccolo episodio, potesse nascere qualcosa di molto innovativo (e anche di molto pericoloso).

Egli immaginò, infatti, lo scenario in cui si potesse svolgere una lotta accanita attraverso i chip di un computer: due programmi, in altre parole, si davano battaglia a colpi di bit, inseguendosi a vicenda nei meandri della memoria, ognuno cercando il punto debole dell'altro e proteggendo il proprio.

Da questa fantasia, poi neanche troppo lontana dalla realtà, scaturì, nel 1983, la "Core War" (Guerra dei nuclei), un vero e

Istruzione	mnemonica	cod.	argomenti		spiegazione
TRASFERIMENTO	MOV	1	A	B	TRASFERISCE CONTEN.DELL'IND.A IN B
SOMMA	ADD	2	A	B	SOMMA I CONTENUTI DELL'IND.A A B
SOTTRAZIONE	SUB	3	A	B	SOTTRAE CONTENUTI DELL'IND.A DA B
SALTO	JMP	4	A		TRASFERISCE IL CONTROLLO ALL'IND.A
SALTO SE ZERO	JMZ	5	A	B	TRASF.CONTR.ALL'IND.A SE B=0
SALTO SE >	JMG	6	A	B	TRASF.CONTR.ALL'IND.A SE B >0
CONFRONTO	CMP	7	A	B	CONFRONTA A E B;SE<>SALTA ISTR.SUCC.
ENUNCIATO DATI	DAT	8		B	UN ENUNCIATO NON ESEGUIBILE

Figura 1

Il termine Core War trae origine dalla terminologia informatica degli anni '50 e '60, quando la memoria dei calcolatori era costituita, appunto, da anelli o nuclei. Si ebbe, quindi, il primo gioco in cui l'uomo si limitava ad assistere, assolutamente impossibilitato ad intervenire durante lo svolgimento, l'attività di due programmi che, come due galli in un'arena, si azzuffavano tra loro per determinare il vincitore.

Il ruolo dell'uomo-programmatore entrava in gioco prima, nel momento in cui il programma doveva passare da idea, embrione, a "concorrente" effettivo: il programmatore, per la stesura del programma, si serviva di un linguaggio speciale, chiamato Redcode, a metà strada tra un linguaggio ad alto livello e quello macchina; una specie di linguaggio assemblatore, per intenderci, basato su proprie regole e istruzioni, ovviamente specializzate.

Da quei primi esperimenti, sollazzo nei momenti di pausa di seri programmatori, siamo arrivati a tornei internazionali dove il vincitore, così come nell'ippica, è dato dall'armonica intesa di programma e autore. proprio genere di gioco del tutto originale.

La storia diventa ancora più avvincente se ci si addentra nell'incredibile logica di funzionamento di questi guerrieri e si capisce come azioni particolari (del tipo piazzare uno sbarramento di bombe o fermarsi per riparare i danni) diventino facilmente traducibili in termini informatici.

ANDARE IN GUERRA

La Guerra dei Nuclei si avvale di quattro componenti principali:

- una matrice di memoria, formata da un certo numero (circa 8000) di indirizzi o locazioni, che funge da campo di battaglia;
- il linguaggio Redcode;
- i due programmi contendenti;
- un programma esecutivo chiamato MARS (Memory Array Redocde Simulator), che riveste le funzioni di arbitro imparziale nel senso che esegue i programmi in maniera alternata (un'istruzione del primo, un'istruzione del secondo) e proclamando, al termine, il vincitore.

Ecco le condizioni preliminari del gioco: l'arbitro ha il compito di porre i due programmi in posizioni casuali della memoria evitando, ovviamente, che vengano a sovrapporsi; ciascun programma ignora la posizione dell'altro; ognuno dei due deve rintracciarlo e attaccarlo, oppure, se preferisce una tattica di difesa, mirare a proteggere le sue postazioni e a ritirarsi sotto il fuoco nemico.

Ma veniamo alla costruzione vera e propria di un programma di guerra.

In figura 1 è riportata la manciata di i-

GUERRE CASALINGHE

Abbiamo pubblicato molto volentieri l'ffascinante argomento proposto dal nostro lettore Fabio Simeoni.

Ed ora, a noi, tocca lanciare la prevedibile sfida:

Siete in grado di realizzare programmi di Core War che possano girare su un Commodore 64?

Non è necessario che siano scritti in linguaggio macchina: si potrebbe, ad esempio, scrivere un codice standard (simile a quello Redcode) e provvedere a trascivere le istruzioni dei due programmi di guerra in altrettanti vettori stringa, ognuno contenente 500 elementi (allocandoli, inutile dirlo, a partire da una posizione casuale). L'effetto delle vicende, gestite da un programma inevitabilmente "imparziale", potrebbe essere evidenziato (interrupt insegna) sulle due aree di 500 byte ciascuna, relative alla memoria video del C/64.

La gara è aperta, si facciano avanti i valorosi guerrieri!

struzioni che costituiscono Redcode (così come è stato creato da Dewdney e Jones).

Molte istruzioni appariranno di certo familiari a chi mastica, anche solo un poco, l'assembler, e, comunque, identica risulta la logica di funzionamento.

Per ciascuna istruzione il programmatore deve fornire uno solo oppure, come nella maggior parte dei casi, due argomenti; questi, se vengono preceduti dal simbolo di cancelletto (#), sono immediati, altrimenti indicano una posizione in matrice.

E' intuitivo che dal momento che il programma non sa in quale posizione della matrice è stato trascritto da Mars, l'unico tipo di indirizzamento eseguibile è quello relativo; ad esempio le istruzioni...

JMP 4

...e...

JMP -4

...indicano a Mars, rispettivamente, di saltare avanti e indietro di quattro istruzioni, e di lì riprendere l'esecuzione del programma.

Quando un argomento viene fatto precedere dal simbolo di chiocciolina, l'indirizzamento diventa indiretto relativo; ad esempio l'istruzione...

MOV (chiocciola) 4 100

...indica a Mars di esaminare il contenuto dell'indirizzo puntato da quattro e di trascriverlo nell'indirizzo 100.

Un'istruzione che richiede maggiori spiegazioni è "DAT".

DAT, che ha un solo argomento, è un semplice enunciato dati e non sortisce alcun effetto; serve solo come spazio di lavoro per il programma (ad esempio, per l'immagazzinamento e successiva ripresa di dati); pertanto non è eseguibili da Mars che, se lo incontra, dichiarerebbe perdente il programma che lo contiene.

Esaminiamo insieme ora alcuni semplici programmi guerrieri scritti da Dewdney e Jones. Il primo programma è Dwarf del quale presentiamo il listato in figura 2.

Dwarf non è un programma molto intelligente ma, in compenso ha un notevole livello di aggressività in quanto bombarda un indirizzo ogni cinque, trascrivendo degli 0, vere e proprie mine per un programma avversario.

Vediamo come: immaginiamo che Dwarf occupi gli indirizzi da 1 a 4 della matrice. La prima istruzione (DAT -1), non sortisce l'esecuzione del programma che, invece, comincia con (ADD #5 -1), aggiunge cioè 5 ai contenuti dell'indirizzo precedente. Questo, di conseguenza, diventa (DAT 4).

La terza istruzione del listato indica a Mars di porre uno 0 nell'indirizzo a cui punta il contenuto dell'istruzione di due posti indietro (cioè DAT 4), quindi di porre uno 0 in posizione 5. Abbiamo quindi visto il nucleo del programma; con questa istruzione la bomba è lanciata, anche se contro un bersaglio non individuato. Con l'ultima istruzione, JMP -2, si crea un ciclo infinito che setaccia, una posizione ogni cinque, lungo l'intera matrice.

```
INDIRIZZO     ISTRUZIONE

0             ....
1             DAT -1
2             ADD #5   -1
3             MOV #0 @ -2
4             JMP -2
5             ....
```

Figura 2: Dwarf

Un programma che si debba difendere da Dwarf potrebbe spostarsi ed eludere il bombardamento, oppure potrebbe subire il colpo e, se dotato della difficile arma della automanutenzione, riparare i danni; oppure, infine, potrebbe cercare di distruggere per primo l'avversario.

LA SOLUZIONE FINALE

Proprio per quest'ultima strategia è nato Imp, un programma guerriero che, se meno intelligente del precedente, è forse il programma più semplice da scrivere in Redcode.

Eccolo:

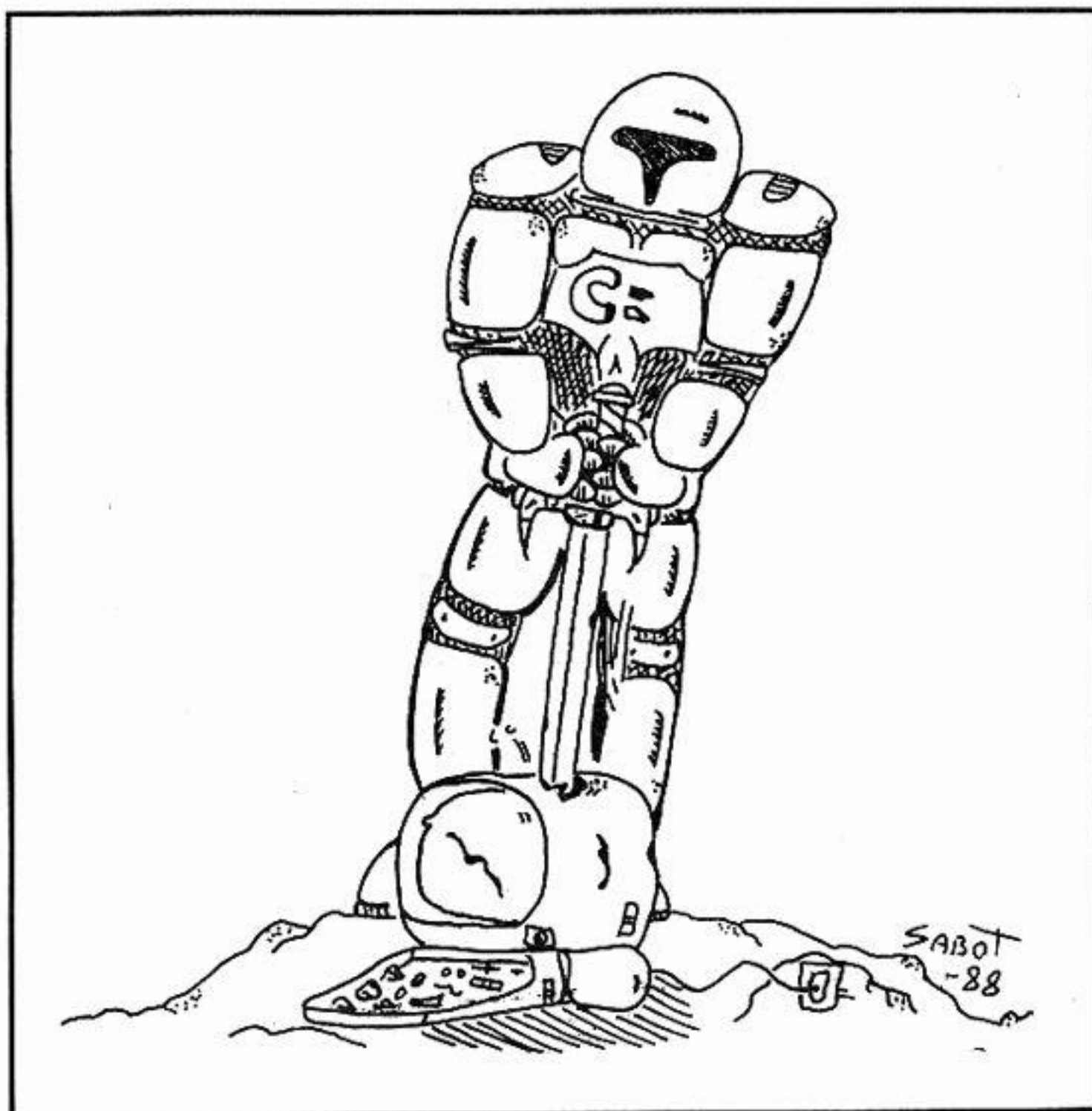
MOV 0 1

Basta quindi una sola istruzione per creare un enorme scompiglio nelle fila avversarie. Imp trascrive i contenuti dell'indirizzo relativo 0 (e quindi lo stesso MOV 0 1), nell'indirizzo relativo 1, cioè quello successivo. In altre parole la sua unica funzione è quella di rigenerarsi una posizione più avanti dell'attuale.

La sua pericolosità sta nel divorare tutto ciò che incontra, lasciandosi dietro una scia di MOV 0 1. Che cosa succede se poniamo nella stessa arena Dwarf e Imp?

Notiamo subito che Dwarf è di certo più veloce di Imp, muovendosi cinque locazioni alla volta, ma esiste la facile probabilità che sia Imp a trovare per primo Dwarf e ad aprirsi un varco tra i suoi codici, cosicchè quando l'istruzione finale di Dwarf JMP -2 trasporterà l'esecuzione indietro di due passi, potrà incontrare un bel MOV 0 1. Dwarf perderà così la sua identità, e, sconvolto, diventerà un secondo Imp che inseguirà il primo all'infinito!

Abbiamo esaminato programmi poco astuti, ma è ovvio che i margini di miglioramento sono pressochè inesistenti; si potrebbe dare più spazio all'aspetto della automanutenzione che renderebbe il programma imbattibile anche se poco aggressivo, magari anche capace di duplicarsi in più copie nella matrice, ognuna delle quali in grado di badare alle altre due per affidare loro il controllo se la situazione si facesse troppo calda.

Insomma è tutto un mondo da scoprire e, anche se le modeste prestazioni di un home computer tarpano le ali alla fantasia, ci si può accontentare anche della schiacciante e affascinante logica di funzionamento dei programmi che si possono progettare.

RESTAURIAMO IL TASTO RESTORE

La gestione del tasto Restore esula dalla norma; con una breve routine, ad esempio...

di **Lorenzo Emilitri**

Avete mai pensato che per la tastiera del vostro amato C/64 si aggiri un infiltrato? Qualcosa che, pur celato sotto le spoglie di un tasto qualsiasi, nasconda una potenza fuori dal comune e tuttora sconosciuta ai più?

I più acuti tra voi l'avranno già intuito... stiamo naturalmente parlando del tasto RESTORE!

Questo non è un tasto comune, quale può essere la "Q", la barra spaziatrice o il tasto Return: è una vera e propria porta che ci può mettere in diretto contatto nientemeno che con il microprocessore in persona!

Stupiti? il bello deve ancora venire!

UN PO' DI STORIA

Narrano le leggende che nei lontani anni '70, quando venne alla luce il microprocessore 6502 (capostipite di una famiglia che avrebbe segnato una svolta nel campo dell'home computing), i suoi progettisti gli diedero parecchie facoltà (paranormali, visti i tempi che correvano...).

Una di esse avrebbe permesso, sempre nelle intenzioni dei suoi creatori, di salvare tutti i dati che aveva elaborato fino al momento in cui si verificava un black-out. La fatidica linea d'ingresso, con priorità assoluta su tutte le altre elaborazioni in corso, era destinata ad essere collegata ad un gruppo statico che alimentasse il computer per tutto il tempo necessario a salvare i dati, e fu battezzata NMI (interrupt non mascherabile).

Al 6502 successe il 6510, suo discendente in linea diretta, ed il Fato volle che proprio questo fosse il prescelto per la nuova creatura della Commodore, il mitico C/64. Il 6510 incorporava alcune migliorie nell'hardware, prima fra tutte una porta di I/O incorporata, pur mantenendo sostanzialmente la stessa impostazione del suo predecessore. E così anche il C/64 fu dotato di una linea NMI.

Visto che l'idea di base del C/64 era quella di una macchina modulare a basso costo, non si potevano certo obbligare i futuri utenti ad acquistare anche un gruppo statico. Così fu scelto un altro impiego per la linea NMI.

Per la sua capacità di prendere il controllo dell'elaborazione in ogni momento (sempre che il processore non fosse in stato di crash), essa fu utilizzata come un metodo di emergenza per riprendere il controllo del calcolatore.

```
100 REM *-----------------------------*
110 REM *      P A U S E  V 1.00      *
120 REM *      BY  - L. EMILITRI -    *
130 REM *-----------------------------*
140 :
150 FOR A = 53200 TO 53241
160 :   READ B : C = C+B
170 :   POKE A,B
180 NEXT A
190 IF C<>5309 THEN PRINT"ERRORE":END
200 POKE 53280,0 : POKE 53281,0
210 PRINT CHR$(147) "PREMI RESTORE PER"
215 PRINT "ENTRARE IN MODO PAUSA"
220 PRINT : PRINT "IN MODO PAUSA:"
230 PRINT " 1) PREMI SPAZIO PER USCIRNE"
240 PRINT " 2) PREMI RUN/STOP PER BREAK"
250 SYS 53200 : END
260 STOP : STOP : STOP : STOP : STOP
270 REM DATA PER LINGUAGGIO MACCHINA
280 REM       (VEDI DISASSEMBLATO)
290 DATA 169,219,141,024,003,169,207
300 DATA 141,025,003,096,072,238,032
310 DATA 208,206,032,208,169,127,141
320 DATA 000,220,173,001,220,201,127
330 DATA 240,007,201,239,240,006,076
340 DATA 220,207,108,022,003,104,064
350 END
```

E così nacque il tasto RESTORE, aiuto spesso insostituibile nell'attività di programmatore, soprattutto quando si traffica con puntatori, set di caratteri, registri video e così via.

A voler essere estremamente precisi, la linea NMI viene utilizzata anche da un altro tipo di dispositivi: le cartucce sprotettrici, che sfruttano la sua capacità di "impadronirsi" del calcolatore per riversare su nastro, o disco, l'intero contenuto della memoria. L'argomento è comunque molto più complesso e delicato di quanto sembri (e, infatti, non lo affronteremo).

LA TEORIA

Quando premiamo il tasto RESTORE (e per qualche ragione ignota, spesso più che di una gentile pressione si deve trattare di un colpo degno dei migliori Wrestlers per convincerlo a collaborare) si genera un interrupt NMI.

Il microprocessore, a questo punto, immagazzina il Program Counter ed il byte di Status nello Stack, e salta ad una routine contenuta nella ROM del Kernel che si occupa di disabilitare gli interrupts; successivamente salta all'indirizzo puntato dalla coppia $0318/9.

A differenza della routine corrispondente, che si occupa degli interrupts IRQ, questa NON immagazzina anche i registri nello Stack, e quindi ciò deve essere fatto dal programmatore.

Per come sono scritte le Rom del C/64, la coppia $0318/9 punta ad una routine Kernel che si occupa di verificare se anche il tasto Run / Stop è premuto; in acso negativo, la pressione di Restore viene ignorata, con conseguente ripresa delle normali elaborazioni. In caso affermativo, invece, viene lanciata una terza routine Kernel (chiamata Break Interrupt e puntata dalla coppia $0316/7), che provoca l'effetto che noi tutti conosciamo quando si preme la coppia Run / Stop + Restore.

Essendo le locazioni $0318/9 allocate in zona RAM, è facile intuire che è sufficiente modificare il loro contenuto (nella consueta forma Byte basso - Byte alto) per inserire una routine personalizzata che girerà in seguito alla pressione del solo tasto Restore.

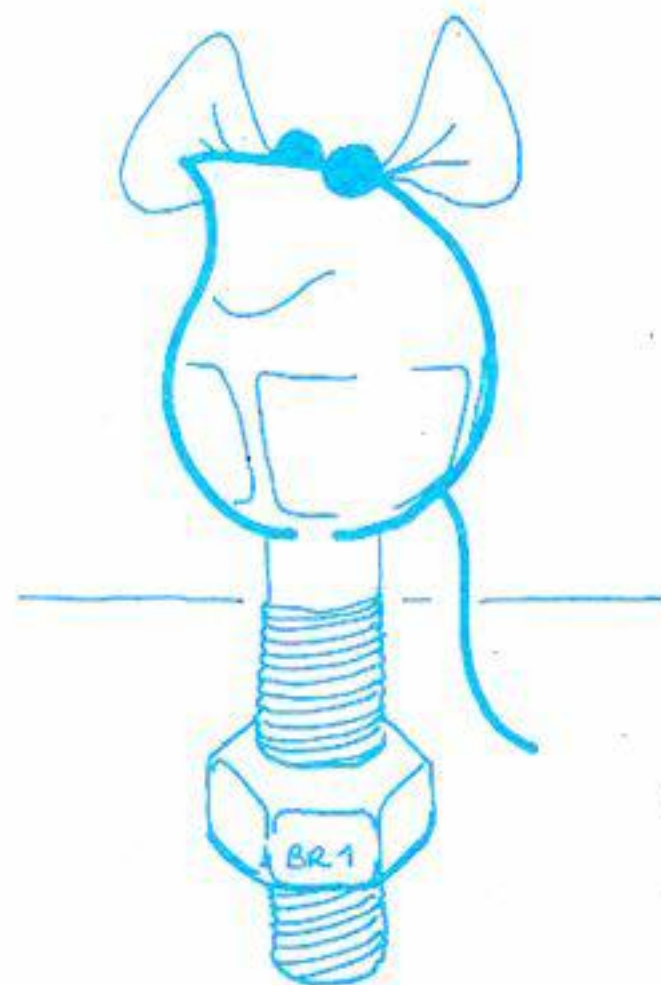
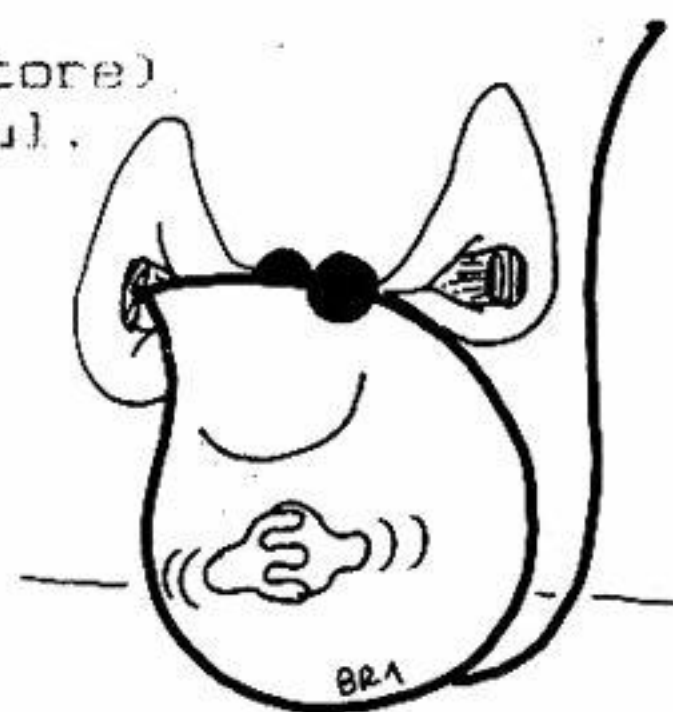
IL PROGRAMMA

Come potete vedere dal disassemblato, il programma presentato in queste pagine serve a "bloccare" il calcolatore, ad esempio mentre sta visualizzando dati per impedire che essi scorrano via; oppure mentre si sta vedendo un programma (anche in Linguaggio Macchina, purchè non sovrascriva la routine di gestione) per inserirvi una pausa anche se ciò non è stato previsto dai suoi programmatori.

Proprio per ottenere la massima compatibilità, la routine non è stata allocata a partire dalla solita locazione 49152 ($C000) ma a partire da 53200 ($CFD0), in modo da permetterne la convivenza pacifica anche con altre routines che utilizzino per scopi propri l'area $C000 - $CFD0.

Una (inevitabile) pecca del programma è che, tentando di realizzare una pausa mentre, ad esempio, si carica la directory di un disco, all'uscita dalla pausa (premendo la barra spazio) il sistema si inchioderà. Il "difetto", riscontrato in tutti i processi che hanno bisogno di temporizzazioni accurate (quali ad esempio la lettura di un file da disco o la stampa di un documento) è da imputarsi all'hardware stesso, poichè l'esecuzione della nostra routine porta ad uno sconvolgimento totale delle delicatissime temporizzazioni.

Proprio per far fronte a questi casi (ed anche per permettere una totale compatibiltà con il consueto uso del tasto Restore) è stato inserito, nel programma, il riconoscimento della pressione di Run / Stop, che causerà l'esecuzione della consueta routine di Break Interrupt.

Premendo Restore (da solo, naturalmente) durante il colloquio con una perife-

```
 1 *----------------------------------*
 2 *        P A U S E    v 1.00       *
 3 *  By Lorenzo Emilitri   110389    *
 4 *----------------------------------*
 5                                     ;
 6            ORG    $CFD0             ;Inizio a 53200
 7                                     ;
 8            LDA    #<NMI             ;Setta il
 9            STA    PNMI              ;puntatore del
10            LDA    #>NMI             ;NMI alla mia
11            STA    PNMI+1            ;routine.
12            RTS                      ;E torna al BASIC
13 *----------------------------------*
14 *  Routine che gira sotto NMI      *
15 *----------------------------------*
16 NMI        PHA                      ;Salva l'accumulatore
17 LOOP       INC    BORDO             ;Disegna lineette
18            DEC    BORDO             ;sul bordo
19            LDA    #$7F              ;Prepara la lettura
20            STA    KBIN              ;della linea #7
21                                     ;della tastiera
22            LDA    KBOUT             ;Legge tasto premuto
23            CMP    #$7F              ;E' Run/Stop?
24            BEQ    BREAK             ;Si', va a BREAK
25            CMP    #$EF              ;E' spazio?
26            BEQ    OUT               ;Si', esce
27            JMP    LOOP              ;Ripete il ciclo.
28 BREAK      JMP    (BRKIRQ)          ;Salta a Break
29                                     ;(= RunStop + Restore)
30 OUT        PLA                      ;Recupera l'accumul.
31            RTI                      ;ed esce.
32 *----------------------------------*
33 *         L A B E L S:             *
34 *----------------------------------*
35 PNMI       EQU    $0318
36 BRKIRQ     EQU    $0316
37 KBIN       EQU    $DC00
38 KBOUT      EQU    $DC01
39 BORDO      EQU    $D020
```

rica, NON premete la barra spaziatrice, ma pigiate Run / Stop.

Un'ultima avvertenza prima di passare alla digitazione del programma: la routine proposta usa la lettura diretta della tastiera (senza alcun aiuto dal Sistema Operativo, ma direttamente dall'hardware): anche questo è un argomento troppo complesso per essere qui trattato, ma vi assicuriamo che se ne parlerà su C.C.C., e più presto di quanto molti di voi credano: continuate a seguirci, ragazzi, e... vedrete!

COMPLICAZIONI

Come ogni buon software che si rispetti, anche questa routine è da considerare come "base" idonea per notevoli sofisticazioni.

Il Demo incluso nel (brevissimo) listato si limita a far lampeggiare il bordo del video quando si preme Restore, dimostrando che, in effetti, è stata attivata una routine diversa dalla solita.

Nulla vieta di effettuare, alla pressione di Restore, la stampa della schermata grafica presente sul video, o comunque un'hard copy di ciò che appare sul vostro monitor: una routine del genere metterà sicuramente a dura prova le vostre capacità di programmatori (riconoscimento del modo grafico attivo, trasformazione da codici video a codici di stampa...) e, magari, vi potrà dare la soddisfazione di apparire su queste stesse pagine!

Buon lavoro, quindi, e attenzione a non ("Aaaargh!") distruggere la vostra povera tastiera!

UN DISCHETTO PIENO ZEPPO DI NOVITA'

Su "AmiGazzetta" n. 5 vi sono ben quindici directory ed 85 files per un totale di oltre 800K di software

Gli Amici di Amiga

Ringraziando i lettori che inviano materiale per la pubblicazione, approfittiamo per ricordare che non interessano programmi macchinosi, sofisticati e che occupano interi dischetti, bensì "cose" semplici ed efficaci, scritti preferibilmente in Amigabasic, C oppure Assembly. Tutto il materiale deve comunque essere inviato sempre su dischetto senza inserire troppi orpelli (non inseribili su AmiGazzetta) e, tassativamente, accompagnato da documentazione in files di notepad, in modo da ridurre i tempi di pubblicazione.

Questo mese AmiGazzetta 5 si è servita del lavoro di Paolo Scalabrini, che ha inviato un simpatico programmino giocoso in AmigaBasic (pubblicato soprattutto perchè illustra bene e semplicemente come usare con tale linguaggio sprites hardware) ed alcune icone per Workbench.

Vediamo ora in dettaglio che cosa ci riserva AmiGazzetta 5.

ICONE

Vi sono 25 nuove icone, pronte per essere usate da Workbench. Alcune sono "animate", cioè cambiano aspetto per indicare la prima attivazione col mouse. Le informa-

zioni sul come gestirle sono state ampiamente descritte su AmiGazzetta, quindi rimandiamo all'indice pubblicato sullo stesso disco con l'elenco di tutti gli articoli apparsi finora su "AmiGazzetta".

AMIGAZZETTA

Per quei pochi che ancora non la conoscessero, AmiGazzetta è una rivista su disco prodotta appositamente per i computer Amiga (500 / 1000 / 2000) in configurazioni standard oppure espanse.

Essa raccoglie programmi di dominio pubblico (tradotti dall'inglese e, comunque, sempre commentati e spiegati in italiano), informazioni utili, icone, disegni, programmi Workbench e CLI, listati sorgenti e compilati in vari linguaggi (C, AmigaBasic, Assembler), realizzata anche con la collaborazione dei lettori.

Ricordiamo che AmiGazzetta è aperta alla collaborazione degli amici lettori di CCC e possessori di Amiga. Se avete programmi in qualunque linguaggio (purchè farina del vostro sacco!), icone, disegni, informazioni che volete divulgare presso tutti gli altri possessori di Amiga, inviate un dischetto (file scritti con un normale WP, icone disegnate con colori standard) al nostro indirizzo.

Ogni numero di "AmiGazzetta" può essere richiesto al prezzo modico di lire 12.000 seguendo le indicazioni riportate in altra parte di queste stesse pagine.

PROGRAMMI CLI

Esistono attualmente in circolazione sei virus per Amiga, gli ultimi dei quali assai subdoli. I vecchi anti-virus non sono più molto efficaci; ecco il motivo della presenza di un utile programma VirusKiller che è in grado di verificare, quando chiamato, tutti i parametri interni che potrebbero (o meglio, devono) essere alterati da un virus per funzionare "correttamente".

Tale programma è in grado di riconoscere tre virus molto noti e di accorgersi, addirittura, della presenza di parametri alterati da eventuali, futuri virus di bootblock (almeno speriamo).

Il programma CLS è un semplicissimo pulitore di schermo, inserito essenzialmente perchè è accompagnato dal testo sorgente in assembler. Vogliamo, infatti, incominciare a dare un'idea su AmiGazzetta di come si programma in linguaggio macchina Amiga (iniziando dagli argomenti più semplici), anche per dimostrare che usare tale linguaggio con Amiga è forse difficile, ma non impossibile. In futuro cercheremo di pubblicare, con la collaborazione dei lettori, altri listati assembly, sempre commentati in italiano.

Il programma BOOT è una sofisticata utility per generare particolari bootblock sui dischetti (ma non virus!). In pratica, si costruisce un particolare file anomalo che genera, quando si inserisce il dischetto alla richiesta del Workbench (dopo un reset completo del sistema), una presentazione coloratissima con una scritta scorrevole fissata dall'utente.

Ciò può essere usato come personalizzazione dei nostri dischetti, per visionare subito (dopo mezzo secondo dall'inserimento) informazioni sul disco, oppure anche come messaggio di presenze di virus (se è presente, il nuovo boot scenografico del disco non parte).

"Timer" è un programma semplice quanto utile. Può essere usato esattamente come un RUN (seguito dal nome di un programma da eseguire e da eventuali parametri) ma effettua un preciso cronometraggio del tempo di esecuzione del programma/comando, con precisione al centesimo di secondo.

PROGRAMMI WORKBENCH

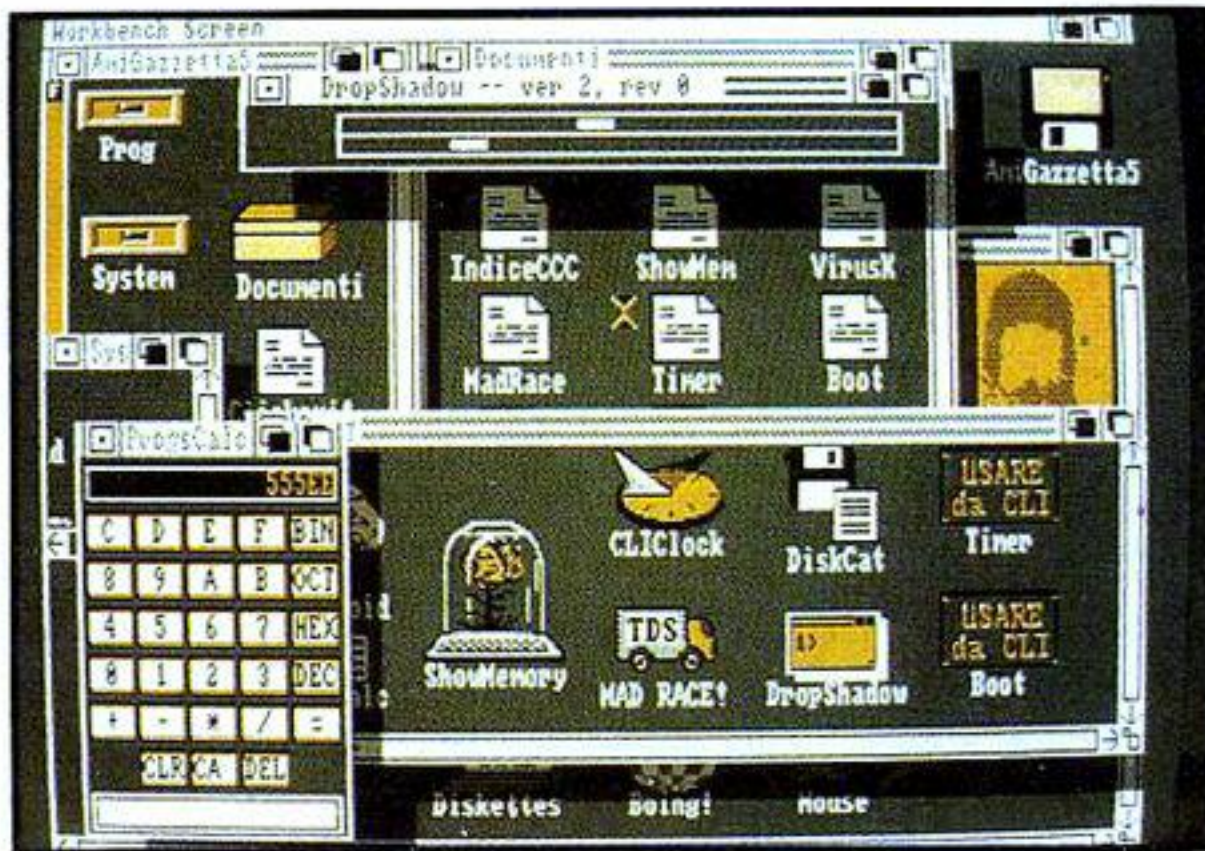
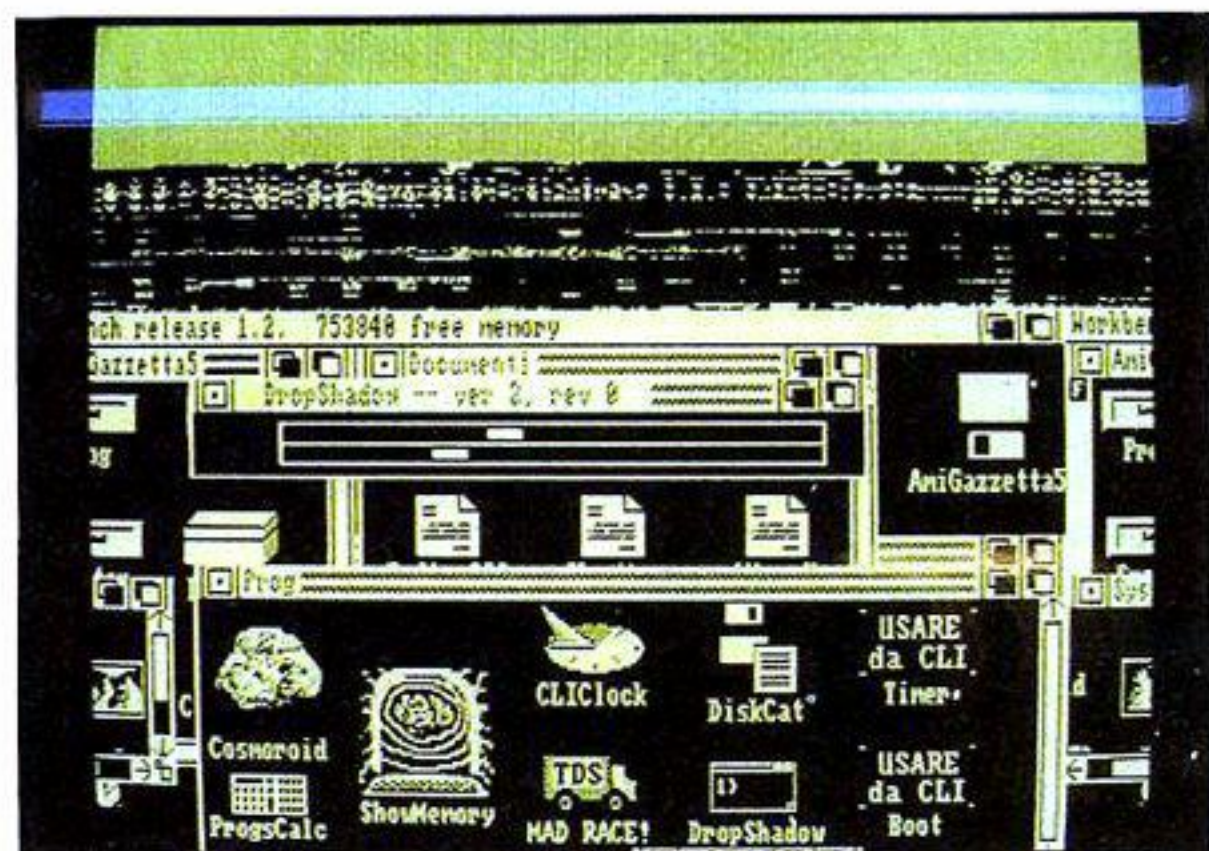
Il programma CliClock è un classico orologino "minimo" che risiede nella barra del menu. Di particolare ed utilissimo ha che clickandoci sopra due volte col tasto destro del topo, in qualunque momento si può aprire una nuova finestra del CLI.

"Cosmoroid" è un divertente e bel videogioco completo, emulo dello storico "Asteroids" Atari. Si controlla una astronave, con joystick o tastiera, in una tempesta di meteoriti da distruggere a suon di laser e di salti nell'iperspazio. Comprende effetti sonori e memorizzazione del punteggio record.

"ProgsCalc" è una nuova calcolatrice per Workbench, questa volta specifica per programmatori in quanto consente calcoli in binario, esadecimale, decimale ed ottale con mouse o tastiera.

"ShowMemory" è un divertente programmino, sempre residente, che mostra il contenuto della memoria di Amiga.

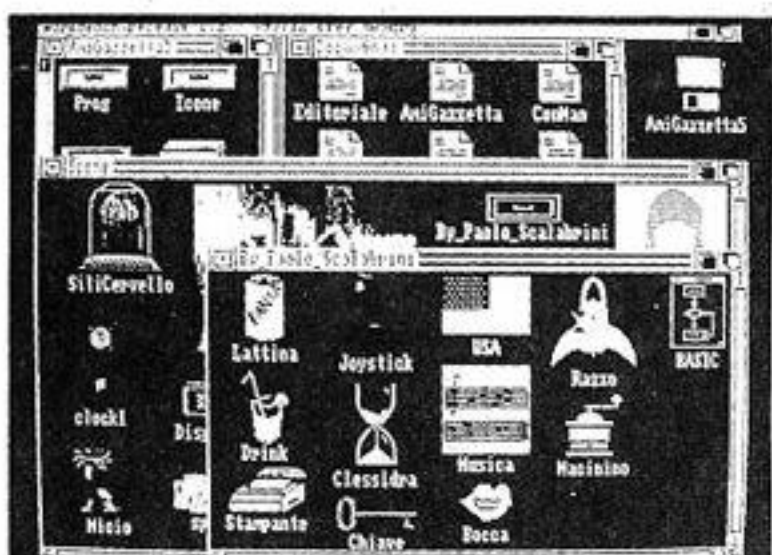
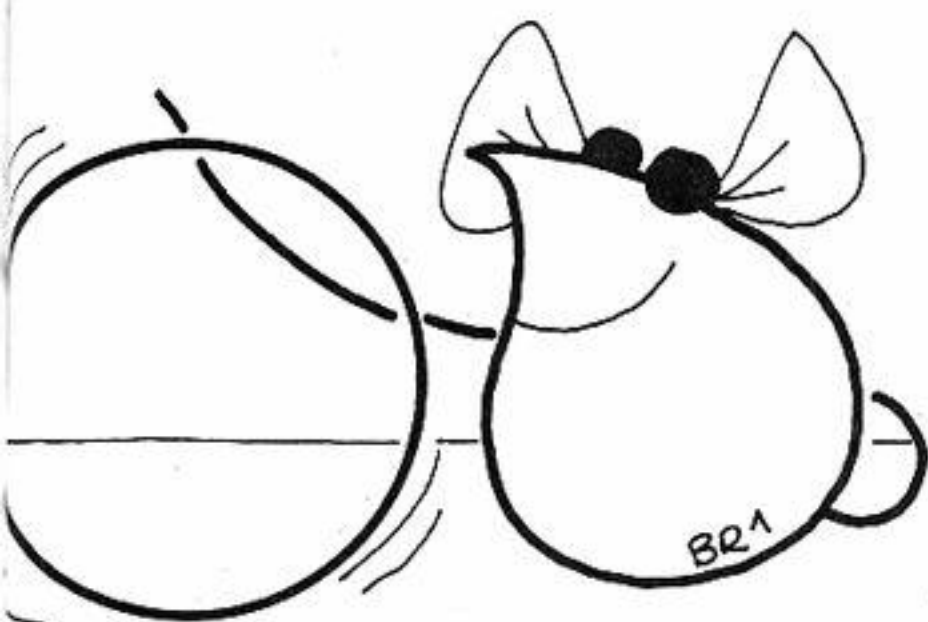
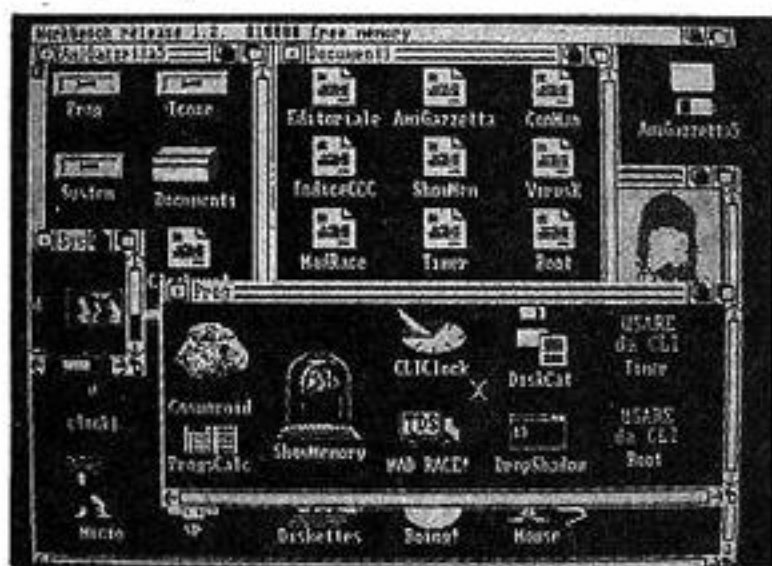
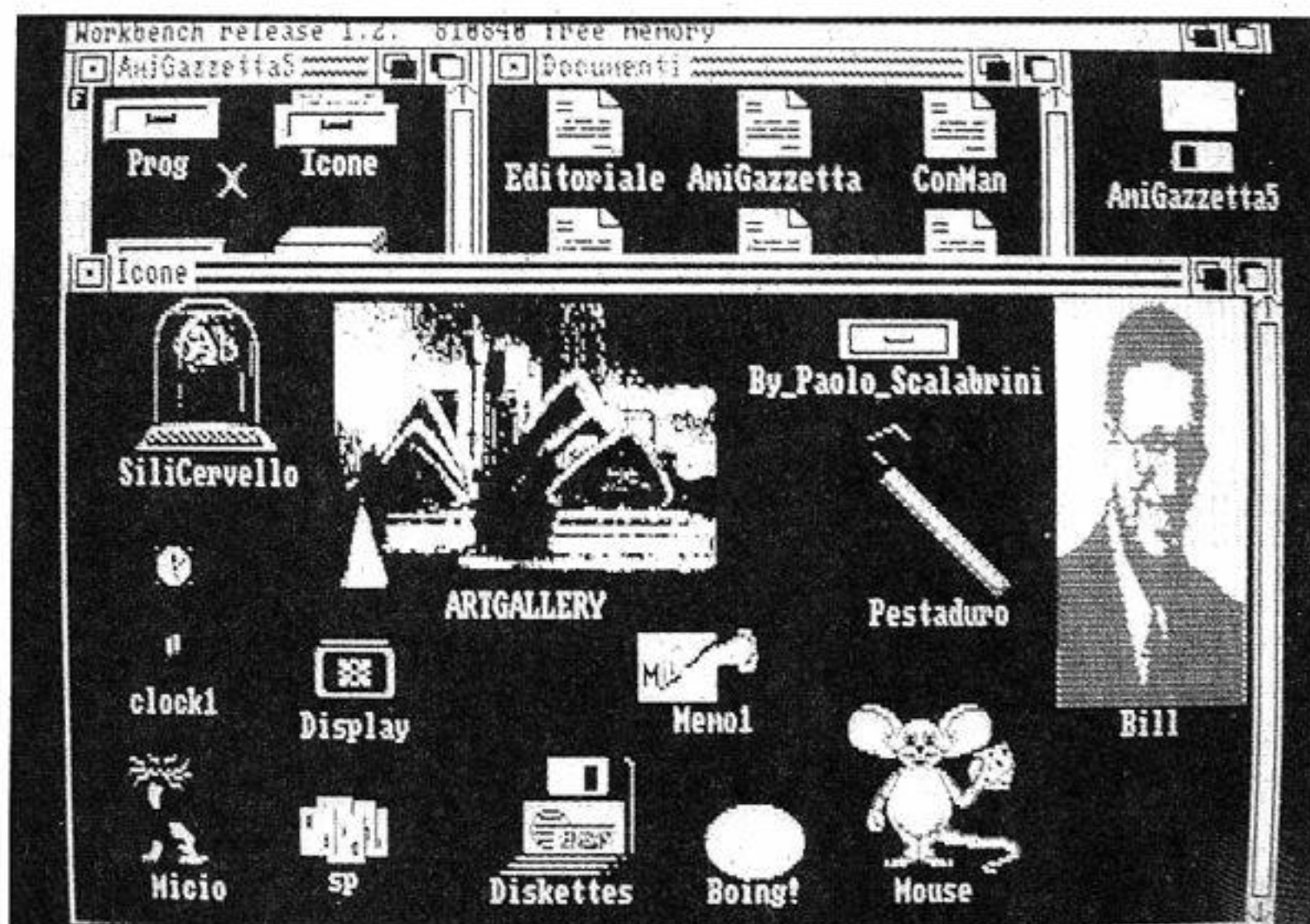
"Mad-Race" è un piccolo programma in AmigaBasic di carattere giocoso, scritto dal lettore Paolo Scalabrini, che illustra bene la gestione di sprites hardware con questo linguaggio.

"Drop-Shadow" è una utility dimostrativa in grado di tratteggiare le ombre di ogni finestra del Workbench, per migliorarne la visibilità e "fare colpo". La densità e la grandezza delle ombre è regolabile con gli slider.

"DiskCat" è un potente archiviatore di dischetti, scritto in C e controllato via mouse e menu. Consente di conservare ordinatamente la propria collezione di dischi, con ordine alfabetico o numerico, suddivisione, ricerca di elementi nonchè stampa su carta.

FILES VARI

Come è tradizione di "AmiGazzetta", ormai, tutti i programmi sono corredati di files di istruzioni completi in lingua italiana.

Inoltre, sono presenti i files di consultazione con l'archivio descrittivo degli articoli apparsi su CCC per Amiga e del software pubblicato su AmiGazzetta.

Speriamo che anche questo numero cinque riscuota il successo dei precedenti e ricordiamo che attendiamo le collaborazioni di tutti i lettori volenterosi.

PER AVERE GLI ARRETRATI

Per richiedere i prodotti Systems che mancano alla vostra collezione è sufficiente inviare al nostro indirizzo, in busta chiusa, un assegno bancario (di qualsiasi Banca, purchè non trasferibile) intestato a "Systems Editoriale - Milano".

La somma da indicare si può determinare riferendosi alle varie pagine informative, rintracciabili in questo stesso fascicolo, relative, ovviamente, ai SOLI prodotti Systems.

Alla cifra così calcolata si deve aggiungere la somma di L. 3000 per le spese di confezione e affrancatura (L. 6000 se si desidera la spedizione raccomandata); la cifra di L. 3000 si riferisce ad ogni spedizione effettuata, indipendentemente dal numero dei prodotti ordinati (dischi, libri, fascicoli arretrati e così via). Si devono aggiungere, insomma, L. 3000 sia per ordinare un solo disco che per ordinare l'intera produzione Systems.

Conviene, pertanto, richiedere più prodotti per volta, mettendosi in società, magari, con qualche amico.

Il lettore non dimentichi di inserire nella busta anche una lettera di accompagnamento in cui siano specificati i prodotti richiesti, oltre al proprio nominativo completo (cognome, nome, indirizzo, c.a.p. telefono).

Il nostro indirizzo è il solito:

Systems Editoriale
Servizio arretrati
Viale Famagosta 75
20142 Milano

Ricordiamo che non ci è assolutamente possibile inviare materiale contrassegno; si prega, quindi, di non insistere.

Se non disponete di un conto in banca, potete inviare la cifra richiesta utilizzando un normale modulo di conto corrente postale intestato a...

C/C N. 37952207
Systems Editoriale
Viale Famagosta 75
20142 Milano

...indicando sul retro del modulo (nello spazio riservato a: "causale del versamento") i prodotti richiesti ed il vostro nominativo completo.

PERO', SI PRESENTA BENE!

Come far apparire, sul video del C/128, una schermata grafica Hi-Res durante il caricamento di nostri programmi

di **Domenico Pavone**

Nell'ambito del software commerciale, quale che sia il computer cui è destinato, è molto diffusa la pratica di inserire, nella fase di caricamento in memoria, immagini spettacolari che fungono da presentazione.

Normalmente si tratta di schermate in alta risoluzione, abbondantemente colorate, e spesso portatrici di messaggi... bucanieri.

Chi non possiede qualche programma che, accanto alla magnificenza di un titolo in caratteri sfolgoranti, segnala un più crudo "cracked by..."?

Hackeraggio a parte, può essere comunque interessante dotare anche i nostri programmmi, almeno quelli migliori, di simili prerogative.

Tanto più, se si possiede un C/128.

Le sue possibilità grafiche, infatti, sono tutt'altro che disprezzabili, e, soprattutto, implementabili con notevole facilità anche da basic.

Diamoci dunque da fare, senza disdegnare qualche sguardo più approfondito all'interno del nostro bel computerone.

PARCHEGGIO RISERVATO

Dopo le roboanti premesse, cominciamo col precisare che prenderemo in considerazione le schermate grafiche multicolor in alta risoluzione.

Il che, com'è noto, significa disporre di uno schermo di 160 x 200 pixel, con la possibilità di sfruttare più di un colore (al contrario della bitmap standard) per ogni singolo punto.

Senza entrare troppo nel merito, si tratta della modalità attivabile, in Basic, con il comando Graphic 3, oppure Graphic 4 se si opta per uno schermo "misto" testo-grafica.

Ma che cosa succede quando si impartisce il comando Graphic 3?

Anzitutto, se in memoria è presente un programma Basic, questo viene trasferito in altra sede (ed esattamente dalla locazione 16384 in poi), per dedicare una porzione di memoria alle informazioni riguardanti la pagina grafica.

Più specificamente, viene riservata, in Ram 0 (bank 0, 14, 15), l'area di 8K compresa tra la locazione 7168 e la 16383.

Quindi, ammesso si sia già creata una immagine in hi-res 160 x 200, sappiamo che questa è disponibile nelle suddette locazioni, pronta per essere usata come "ingrediente" per i nostri scopi.

Per essere completa, però, manca ancora di un dato indispensabile: l'informazione sui colori adoperati.

Questi, nel caso del multicolor, sono memorizzati anche in un'altra zona (di 1 KRam), la ben nota fascia di memoria che va da 55296 a 56319 di banco 15, usata anche in modo testo (graphic 0) per "pokarvi" il colore corrispondente ad ogni cella di schermo.

Anche qui tutto sembra semplice, e potremmo considerare ultimata la ricerca degli elementi utili alla nostra schermata di presentazione.

Ma, ovviamente, non è così.

DUE MEMORIE IN UNA

Un po' di pratica alla tastiera chiarirà l'aspetto del problema.

NON SOLO MULTICOLOR

L'uso della grafica con risoluzione 160 x 200 pixel, pur consentendo notevoli risultati, grazie soprattutto alle maggiori possibilità cromatiche, non sempre risulta all'altezza della situazione.

Si può, infatti, desiderare una migliore definizione dei particolari, o (potrbbe essere il caso degli esempi proposti nell'articolo), una più chiara leggibilità di eventuali caratteri inseriti nella schermata.

In questi casi, nulla vieta di ricorrere alla massima risoluzione implementabile su 40 colonne, 320 x 200 pixel.

Occorre, ovviamente, prestare maggiore attenzione nell'accostare colori che possono "impastarsi", ma anche Graphic 1 può dare ottime soddisfazioni.

Da un punto di vista pratico, la tecnica da adottare per visualizzare una immagine in hi-res "standard" durante il caricamento di un altro programma, per certi versi, risulta ancora più facile che in multicolor.

A parte l'ovvio uso di Graphic 1 al posto di Graphic 3, la differenza che interessa riguarda la mappa colore, memorizzata nell'area da 55296 in poi, ma nelle 1000 locazioni che costituiscono la matrice video.

Questa, in modo testo, è letta dal sistema a partire dall'indirizzo 1024; dopo un comando Graphic 1, invece, attraverso una manipolazione dei registri del VIC (automatica, la cosa non ci interessa direttamente), la stessa viene trasferita a partire da 7168, ovvero all'interno della porzione di memoria riservata all'hi-res.

Si badi che non viene trasferita "fisicamente": le locazioni da 1024 in poi rimangono intatte, per cui lo schermo normale non subisce interferenze.

Da tutto ciò se ne ricava che, al momento di salvare su disco la schermata, sarà sufficiente prendere in considerazione un solo file, contenente la suddetta area da 7168 a 16383 (di Ram 0).

La differenza può essere apprezzata meglio rifacendosi ai listati 3 e 4, che ripropongono la tecnica già affrontata a proposito del multicolor, adattandola alle esigenze di una grafica meno variopinta, ma più dettagliata.

Si resetti il sistema, quindi si scriva qualcosa sulla prima riga dello schermo.

Portandosi, con il cursore, più in basso di un paio di righe, si digiti ora:

20 for x=0 to 39: poke 55296+x, 1: next

Impartendo il Run, tutti i caratteri presenti in cima allo schermo si coloreranno di bianco, come era ovvio aspettarsi.

Si aggiunga ora, a quella appena digitata, quest'altra linea Basic...

10 poke 1, peek(1) and 254

...che, essendo numerata con 10, sarà eseguita per prima.

Si premano Run/stop + Restore, si scriva qualcos'altro sulla prima riga dello schermo, quindi si dia di nuovo Run: stavolta, non accade (apparentemente) niente.

Per rimettere le cose a posto (riprovare per credere), impartiamo ora, direttamente:

poke 1, peek(1) or 1

...e Return. Cos'è successo?

Semplice: il sistema, ovvero il VIC, va a cercare i colori nell'area da 55296 in poi, dove noi vi abbiamo "pokato" il bianco, ma non lo trova, nè può trovarlo.

Il fatto è che, agli stessi indirizzi da 55296 a 56319, sono presenti DUE aree di memoria, non una soltanto, similmente a quanto avviene sul C/64 per la Ram "nascosta" sotto l'interprete del Basic.

Qui, però, si tratta in entrambi i casi di Ram, sempre nel contesto del banco 15, che possiamo immaginare come "sovrapposte": una, detta blocco 0, viene adoperata per memorizzarvi i colori dell'alta risoluzione multicolor, l'altra (blocco 1), serve alla comune gestione dello schermo-testo.

Per selezionare una delle due, come forse avrete capito, occorre modificare la locazione 1 (altra somiglianza col c/64), con le poke appena viste.

Per i più esperti, va detto che, in funzione del multicolor, va manipolato il bit 0: settato, farà riferimento all'area testo, azzerato, alla bitmap.

In effetti entra in gioco anche il bit 1, ma per i nostri scopi può essere ignorato; è sufficiente ricorrere ai comuni comandi Basic che selezionano il tipo di grafica attiva.

UN LOAD PIU' VIVACE

Bene, finalmente disponiamo di tutto il necessario... teorico per realizzare la nostra schermata, da esibire durante il caricamento di un altro programma.

Non resta, dunque, che passare alla pratica, riassumendo i passi da compiere per raggiungere lo scopo:

1) Preparare un'immagine in hi-res.

2) Memorizzare sul disco (non avrete mica il registratore soltanto!!??) contenente il programma da caricare, l'area di memoria riservata all'hi-res.

3) Trasferire sullo stesso dischetto anche la mappa colore, dopo avere selezionato quella utile (il blocco 0).

4) Preparare un "loader" che provveda prima a caricare, e mostrare, la schermata con i suoi colori, e successivamente il programma desiderato.

Più concretamente, si cominci con il copiare (e salvare) il listato 1 di queste pagine, che provvede a realizzare quanto precisato ai punti 1, 2 e 3.

Esaminiamolo meglio.

Il primo blocco di comandi, le righe da 140 a 230, si limitano a creare un... improbabile messaggio di copyright, operando in hi-res multicolor (graphic 3).

Chiaramente, sarà vostra cura preparare qualcosa di più interessante, magari sfruttando, se ne siete in grado, qualche "tool" grafico tipo Koala painter.

Ultimata la schermata, con riga 250 e 260 si prelevano, dagli adeguati registri (più che noti), il colore dello sfondo e quello del bordo, per depositarli nelle due locazioni immediatamente precedenti l'area riservata alla grafica, ovvero la 7166 e la 7167.

Questo consentirà, con un'unica istruzione Bsave, di salvare tutta l'area da 7166 a 16383 (riga 270), che occuperà 37 blocchi del dischetto.

Infine (righe 280 - 300), si seleziona la ram colore per la bitmap, la si salva con un'altra Bsave, e si ripristinano le condizioni di default.

Tutto fatto.

Ora, sul dischetto, saranno presenti due file di nome "bitmap" e "colori", oltre, ovviamente, il programma che si desidera lanciare dopo la visualizzazione della schermata.

Non resta che memorizzarvi un'ultimo file, il listato 2.

Si resetti dunque il computer e si copi il listato, salvandolo poi, con un nome tipo "loader", nello stesso dischetto contenente i file "bitmap", "colori" e il programma da lanciare, magari il "Demo" proposto in queste pagine.

Il loader si limita a caricare la bitmap e, dopo aver settato opportunamente la locazione 1, anche la mappa colore (righe

140 - 170).

Dopodichè, regola il bordo e lo sfondo adoperando il contenuto delle locazioni 7166 e 7167 (le avevamo salvate assieme alla bitmap, ricordate?), e con un semplice Dload carica il programma da lanciare, il "Demo".

Quest'ultimo, un semplice esempio, contiene un ciclo di ritardo per ovviare alla rapidità con cui viene caricato, dopodichè, alla pressione di un tasto, torna in graphic 0 ed effettua un List, a comprova della sua esistenza.

Il comando Graphic 0, è pressochè obbligatorio (anche nei vostri elaborati), almeno se si utilizza il loader per programmi Basic.

In questo caso, infatti, il Dload, usato non in modo diretto, non solo provvede al caricamento, ma anche al lancio (automatico) con Run.

Quindi, se si intende tornare a normali mansioni grafiche, è necessario che se ne occupi il programma eseguito per ultimo.

La stessa tecnica, comunque, è applicabile anche a programmi in linguaggio macchina, semplicemente inserendo un Bload al posto di Dload, e aggiungendo una riga con la Sys di attivazione.

In questo caso, un eventuale Graphic 0 può essere posto direttamente nel loader, prima della Sys.

Si noti che il comando Fast (riga 130), è applicato non tanto per velocizzare il caricamento, che praticamente non ne viene influenzato, quanto per il "blank" dello schermo che ne consegue.

Lo stesso effetto, in pratica, si otterrebbe azzerando il bit 4 del registro 53265, ma, piuttosto che ammattire con And e Or vari (il discorso non vale per gli "Assemblysti"), è molto più immediato ricorrere a Fast e Slow.

Senza azzerare lo schermo, si vedrebbe tutto il lavoro di "costruzione" dell'immagine e inserimento dei colori, con un effetto che, comunque, potrebbe anche trovare i suoi estimatori: togliete il Fast, e giudicate da soli.

ALLA FINE

Bene, siamo in dirittura d'arrivo, e ormai dovrebbe essere chiaro che, per visualizzare una schermata grafica durante il caricamento di un programma, basta disporre, sullo stesso dischetto, del programma, della bitmap, della mappa colore, e del loader.

Se tutto è in regola, l'unica operazione da compiere è il lancio di quest'ultimo.

Il modo migliore per sfruttare questa tecnica, è intuitivo, consiste nell'inserire il loader nell' autoboot di un dischetto, sfruttando le utility già da tempo in circolazione, o provvedendo da soli.

```
100 REM ----------------------
110 REM          LISTATO  1
120 REM ----------------------
130 :
140 BANK15:F1$="COPYRIGHT BY"
150 F2$=" ALBERT EINSTEIN"
160 COLOR0,1:COLOR4,1:COLOR1,3
170 COLOR2,2:COLOR3,7:GRAPHIC3,1
180 CIRCLE2,80,100,70,33
190 CIRCLE2,80,100,74,37
200 CHAR2,6,12,F1$+F2$,0
210 BOX2,2,50,158,152
220 PAINT1,54,100,1
230 PAINT3,4,54 ,1
240 REM ----------------------
250 POKE7166,PEEK(53280)
260 POKE7167,PEEK(53281)
270 BSAVE"BITMAP",P7166TOP16383
280 POKE1,PEEK(1)AND254
290 BSAVE"COLORI",P55296TOP56319
300 POKE1,PEEK(1)OR1:GRAPHIC0
310 END
```

```
100 REM --------------------
110 REM      LISTATO  2
120 REM --------------------
122 REM QUESTO LISTATO PUO'
123 REM ESSERE ATTIVATO SOLO
124 REM SE, IN PRECEDENZA, E'
125 REM STATO ATTIVATO IL
126 REM LISTATO N.1
127 :
130 FAST:BANK15
140 GRAPHIC3,1:BLOAD"BITMAP"
150 POKE1,PEEK(1)AND254
160 BLOAD"COLORI"
170 POKE1,PEEK(1)OR1
180 POKE53280,PEEK(7166)
190 POKE53281,PEEK(7167)
200 SLOW
210 REM --------------------
220 DLOAD"DEMO"
225 :
230 REM E' INDISPENSABILE
240 REM CHE SUL DISCHETTO
250 REM SIA PRESENTE IL FILE
260 REM DI NOME "DEMO"
```

```
100 REM ----------------------
110 REM          LISTATO  3
120 REM ----------------------
130 :
140 BANK15:F1$="COPYRIGHT BY"
150 F2$=" ALBERT EINSTEIN"
160 COLOR0,5:COLOR4,5:COLOR1,1
170 GRAPHIC1,1
180 CIRCLE1,160,100,140,33
190 CIRCLE1,160,100,148,37
200 BOX1,2,50,316,152
210 PAINT1,4,54,1:COLOR1,2
220 CHAR1,6,12,F1$+F2$,0
230 REM ----------------------
240 POKE7166,PEEK(53280)
250 POKE7167,PEEK(53281)
260 BSAVE"BITMAP2",P7166TOP16383
270 GRAPHIC0:END
```

```
100 REM ------------------
110 REM       LISTATO  4
120 REM ------------------
125 :
130 FAST:BANK15:GRAPHIC1,1
140 BLOAD"BITMAP2"
180 POKE53280,PEEK(7166)
190 POKE53281,PEEK(7167)
200 SLOW
210 REM ------------------
220 DLOAD"DEMO"
```

```
10 REM ---------------
20 REM       DEMO
30 REM ---------------
40 A$="PREMI UN TASTO"
50 FORX=1TO1500:NEXT
70 CHAR1,12,24,A$,1
80 GETKEYB$:GRAPHIC0
90 SCNCLR:LIST:END
```

CAMPUS

LABORATORIO SOFTWARE DI COMMODORE COMPUTER CLUB

PRINCIPIANTI

- Certo, certissimo, anzi probabile
- Equazioni differenziali di ordine Primo e Secondo

ESPERTI

- Una schermata tutta da tradurre
- Dio salvi gli Sprite

CAMPIONI

- K-Seka Assembler, primo incontro

CERTO, CERTISSIMO, ANZI: PROBABILE

In clima di elezioni è opportuno sperimentare alcune curiosità matematiche; ed i metodi per evitar sorprese

di **Alessandro de Simone**

Non è la prima volta che viene affrontata, su queste pagine, la Statistica, magari in forma "leggera" e divertente.

Stavolta vedremo i ragionamenti da seguire per effettuare ricerche di vario tipo su eventi che stanno per manifestarsi.

Accenneremo, lo avete capito, alle famose "proiezioni" di cui tanto sentirete parlare in questo periodo.

Quando si desidera conoscere la conclusione di un evento, e non si desidera (o non si può) attendere che questo si concluda, si ricofre a determinazioni probabilistiche che, in un modo o nell'altro, riescono a "prevedere" (termine forse presuntuoso) l'evento stesso, in tutti i suoi dettagli.

A questi sistemi, la cui struttura varia moltissimo a seconda dei casi (e di coloro che la realizzano) ricorrono le più svariate categorie di persone: dal giocatore del lotto, che si basa sui famigerati "ritardi", ai giocatori del Totocalcio, che non solo attribuiscono validità più o meno accettabili alle varie squadre in gioco, ma determinano in modo altrettanto personale il numero delle colonne da giocare.

Un po' più seriamente dovrebbe, però, essere affrontato uno studio del genere, soprattutto quando in ballo ci sono molti quattrini (oppure il potere, che è la stessa cosa...).

Rinviando il lettore al riquadro per ulteriori notizie, vediamo di esaminare il programma proposto ed il modo di utilizzarlo.

IL PROBLEMA

Vi sono diversi dati sui quali è possibile arzigogolare.

Una prima categoria appartiene ad un evento che non si conosce affatto, ma che ha molte probabilità che si verifichi:

Se, ad esempio, al 30' del secondo tempo di una partita di pallone la squadra "A" sta vincendo per 8 a 1, è molto probabile che vinca definitivamente la partita; nessuno, però, può dirlo con certezza nè, tantomeno, era possibile prevedere il risultato quando, al primo minuto del primo tempo, si registrava un semplice 0-0!.

Mentre, però, nel caso appena esaminato si può prevedere la vittoria della squadra "A" con una certa tranquillità, vi sono situazioni che sono già "fisse" ed immutabili e ci si può avvicinare alla loro determinazione un po' per volta.

Nel caso delle elezioni, ad esempio, quando inizia lo spoglio delle schede, il risultato è già "esistente" e non può mutare a seconda dell'umore dei socialisti o delle speranze dei comunisti: il voto segnato su ciascuna scheda è indipendente da qualsiasi fenomeno avvenuto DOPO l'inserimento della scheda stessa nell'urna da parte dell'elettore.

Si tratta, quindi, di prevedere, a mano a mano che procede lo spoglio, la ripartizione TOTALE di tutti i voti. La percentuale esatta sarà possibile esclusivamente dopo l'estrazione dell'ultima scheda.

Nel caso delle elezioni, quindi, sarebbe sufficiente attendere 24 ore dalla chiusura dei seggi per avere i dati ufficiali. Ma, si sa, siamo ormai abituati alla suspence, ai personaggi intervistati durante le lunghissime e noiose attese tra un dato e l'altro.

Vi sono casi, tuttavia, in cui può essere utile limitarsi a determinare alcune percentuali, pur se approssimate (ma sarebe megio definirle "orientative"), relative all'evento studiato.

Sei in grado di sofisticare, da solo, il programma pubblicato?

LE AVVENTURE DI

PRIMO GIOVEDINI

by Marco Mietta
ha collaborato
Barbara De Toffoli

Un serio "inchiodamento". (1° File)

AUDITEL E IL POPOLO

Tutti noi abbiamo assistito, almeno una volta, alle battaglie combattute sul fronte della TV. L'audience, infatti, è il nuovo Dio che regola, ed impone, il tipo di spettacolo da propinare ai teleutenti.

Evitando falsi ed inopportuni pudori, è universalmente risaputo che l'arma "Televisione" (perchè di arma si tratta, più micidiale di quanto non si voglia ammettere) vive grazie al denaro che è in grado di far arrivare nelle casse dell'Ente (pubblico o privato) che la gestisce.

La differenza, infatti, è sottile: se l'emittente TV è "commerciale", si tratta di denaro; se, invece, è di altro tipo, si tratta di "potere". L'uno o l'altro si ottiene grazie all'audience, vale a dire la capacità di accalappiare spettatori e di convincerli che le idee trasmesse sono proprio ciò che di meglio si può pretendere.

In nome dell'audience, ad esempio, si stabilisce se un omicidio in diretta attira di più o di meno di una persona ritrovata, anche se in differita. Atteggiamenti moralistici vengono subito messi da parte grazie alla classica, disarmante obiezione degli addetti ai lavori:

"A che serve trasmettere programmi educativi se nessuno li guarda?".

Limitando le polemiche, e tornando a noi, la guerra dell'audience si combatte, in Italia, sul campo dell'Auditel. In pratica si tratta di qualche migliaio di famiglie (i cui nominativi sono tenuti segreti per ovvi motivi) che hanno accettato di collegare il proprio apparecchio TV ad uno speciale scatolotto elettronico, collegato permanentemente con un centro di elaborazione dati, che fornisce informazioni sullo "stato" dell'apparecchio stesso: se è acceso o spento, su quale canale è sintonizzato, per quanto tempo rimane sintonizzato su una certa emittente e così via.

Al temine della giornata, oltre che istante per istante, si effettua una statistica e si determina il numero di apparecchi sintonizzati su una frequenza, lo si paragona con gli altri e lo si estrapola. Con quest'ultimo termine si intende quell'operazione grazie alla quale è possibile sapere quante persone, in tutt'Italia, hanno visto una certa trasmissione.

Supponiamo, per semplicità, che gli apparecchi TV collegati con Auditel siano 1000 e che invece, in tutt'Italia, siano in realtà 10 milioni. Ogni apparecchio Auditel, quindi, "rappresenta" 10000 apparecchi e, di conseguenza, altrattante unità familiari. Se, quindi, 300 apparecchi Auditel vedono Pippo Baudo, si deduce che 3 milioni di italiani hanno preferito vedere il programma nazional popolare piuttosto che altre trasmissioni. Chissà che c'era sulle altre reti in quel momento!

Scherzi a parte, il calcolo è attendibile se i 1000 utenti Auditel sono realmente rappresentativi della popolazione. Sarebbe un errore selezionarli tutti nelle città (trascurando la provincia) o solo tra i ricchi (trascurando i meno abbienti), o solo tra gli anziani (trascurando i giovani) e così via.

L'ideale, è ovvio, sarebbe collegare OGNI apparecchio TV con Auditel, ma ciò, fortunatamente, non è possibile.

Supponiamo, ad esempio, che venga svolta un'inchiesta tra i lettori di una rivista e che giungano in Redazione, contro le aspettative, 10000 schede invece delle previste 5000. Bisognerebbe, per rigore logico, esaminare le 10000 schede prima di trarre conclusioni statistiche. Invece non è così.

La differenza esistente tra una statistica condotta su 5000 schede, o su 10000, è talmente piccola che non compensa il doppio del lavoro occorso per "inserirne" 10000 nel computer.

Altro esempio, pur se banale: sappiamo tutti che, avendo a disposizione un dado non truccato, ognuna delle sei facce ha la stessa probabilità di estrazione. Effettuando 600 lanci, insomma, ogni faccia dovrebbe comparire 100 volte. La precisione aumenta se, invece, di 600 lanci, ne effettuiamo

%= .5	140	160	190	470
228	48.57	48.75	48.34	48.51
242	51.42	51.25	51.05	51.48
48.6	.22	.37	.38	-.19
51.3	-.22	-.34	-.4	.17

Figura 1

%= 1	60	160	190	430
209	51.66	48.75	48.94	48.6
221	48.33	51.25	51.05	51.39
48.2	-.66	.37	.38	.82
51.7	.68	-.34	-.4	-.78

Figura 2

%= 5	50	70	160	200
99	52	54.28	48.75	49.5
101	48	45.71	51.25	50.5
48.7	3.99	4.38	.59	1.53
51.2	-4	-4.78	-.55	-1.47

Figura 3

%= 10	50	70	100	140
68	52	54.28	51	48.57
72	48	45.71	49	51.42
51	3.99	4.38	-6.05	-4.77
49	-4	-4.78	7.19	4.93

Figura 4

%= 20	20	40	70	110
57	55	50	54.28	51.81
53	45	50	45.71	48.18
54.2	10	-9.1	8.56	-4.56
45.7	-10	11.11	-8.59	5.4

Figura 5

6000, 60000 e così via. Se avessimo il tempo di farli, ci accorgeremmo, però, che le percentuali rilevate si differenziano tra loro di pochissimo, almeno a partire da un certo numero di lanci in poi.

Allo stesso modo (e, lo ripetiamo, volendosi accontentare di dati "orientativi" e non di dati esatti) è inutile attendere l'estrazione dell'ultima scheda dall'ultima urna per stabilire la vittoria di una partito o la sua sconfitta (però, in questi casi, si parla di "flessione", se non di "tenuta", chissà poi perchè...).

Determinare le percentuali è molto importante quando si parla di statistica

DAL PROBLEMA AL COMPUTER

Uno studio sulla attendibilità di un evento sembrerebbe di semplice attuazione con un computer domestico.

C'è da dire, però, che non è così semplice e non solo per questioni di linguaggio, ma proprio come impostazione del problema.

Supponiamo di volere esaminare un caso banale, per quanto insolito: che probabilità abbiamo che, generando un numero casuale compreso tra 1 e 2000, questo capiti negli intervalli 1/1000 e 1001/2000? Sembrerebbe, se il computer non bara, che la probabilità sia del 50% in entrambi i casi. Altrettanto ovvia dovrebbe essere la considerazione secondo cui maggiore è il numero di estrazioni, maggiore è la possibilità che si verifichi la situazione ottimale.

Quante estrazioni, quindi, è sufficiente effettuare per trarre conclusioni attendibili e come possiamo avere una conferma che il numero minimo di estrazioni è quello richiesto?

Abbiamo infatti affermato che il computer non bara, ma chi può dire se un calcolatore non ha particolari predilezioni per un certo intervallo numerico e non per altri?

Per rispondere a questa domanda bisognerebbe effettuare dapprima 1000 estrazioni, poi 2000, poi 3000. Se le differenze dalle aspettative sono giudicate eccessive, continuare con 4000, 5000 e così via.

Il metodo, però, lungo e noioso, porta spesso a risultati inaspettati: il risultato che si ottiene dopo 100 mila estrazioni può essere eguale a quello che si era ottenuto dopo appena 100 estrazioni! Ciò dimostra che non sempre, aumentando il numero delle estrazioni, aumenta la precisione delle previ-

Alcuni algoritmi possono essere adattati alle varie necessità che si presentano

sioni. Anzi: con il programma di queste pagine ci siamo accorti che un numero eccessivo di estrazioni può portare a risultati poco attendibili.

IL PROGRAMMA

Il listato di queste pagine sfrutta in modo intensivo l'istruzione RND che, nel linguaggio Basic, ha una singolare e comoda particolarità: se (riga 150) si genera un numero casuale dotando la funzione RND di argomento NEGATIVO, a partire da quel momento il computer genererà una sequenza di numeri che sarà sempre la stessa tutte le volte che il programma verrà fatto partire. Anche spegnendo e riaccendendo il computer (e ricaricando il programma) si otterranno gli stessi risultati. Se non ci credete, provate a digitare il seguente programmino...

```
10 A = Rnd(-5)
20 Print Rnd(1): Goto 20
```

...e verificate che tutte le volte che lo fate girare (con Run) viene visualizzata la stessa sequenza di numeri.

Volendo ottenere numeri realmente casuali sarà sufficiente cancellare la riga 150 e modificare la 420 (Rnd (0)).

Ottenere sempre la stessa sequenza sarà utilissimo, nel nostro programma, per verificare la bontà di un metodo rispetto ad un altro. Se, infatti, due elaborazioni forniscono risultati diversi, ciò potrebbe essere attribuito alla casualità dei numeri trattati dal metodo stesso. Avendo, al contrario, a disposizione sempre gli stessi numeri, la differenza dovrà essere attribuita esclusivamente alla differente elaborazione effettuata.

All'inizio il programma chiede il numero di "elementi" (classi) alle quali dovrà appartenere il numero da estrarre. Ciò significa, ad esempio, che se rispondete con "3" il computer chiederà, per ciascuna di queste, il valore massimo attribuibile a ciascun intervallo.

Spieghiamoci meglio con un esempio:

N. Elementi? 3
Range Max. elem 1? 40
Range Max. elem 2? 93
Range Max. elem 3? 118

Anzitutto c'è da dire che, ovviamente, ogni valore deve esser maggiore del precedente (riga 190); inoltre il valore massimo che il computer potrà estrarre sarà proprio l'ultimo digitato (riga 270) che, nel nostro caso, è 118.

A partire da questo momento verranno generati numeri casuali interi (riga 420) compresi tra 1 e 118. Una volta estratto, il numero incrementerà il "contatore" del primo, secondo o terzo elemento a seconda se il numero stesso risulta compreso, rispettivamente, tra 0 e 40, tra 41 e 92, tra 93 e 118.

A questo punto verranno effettuate 10 estrazioni (variabile T, riga 240 e 260) e verranno incrementati, e visualizzati, i corrispondenti contatori di appartenenza. Al termine verrà effettuata la statistica e compariranno, al fianco dei contatori, le relative percentuali (il loro totale, a meno di arrotondamenti, è infatti pari sempre a 100).

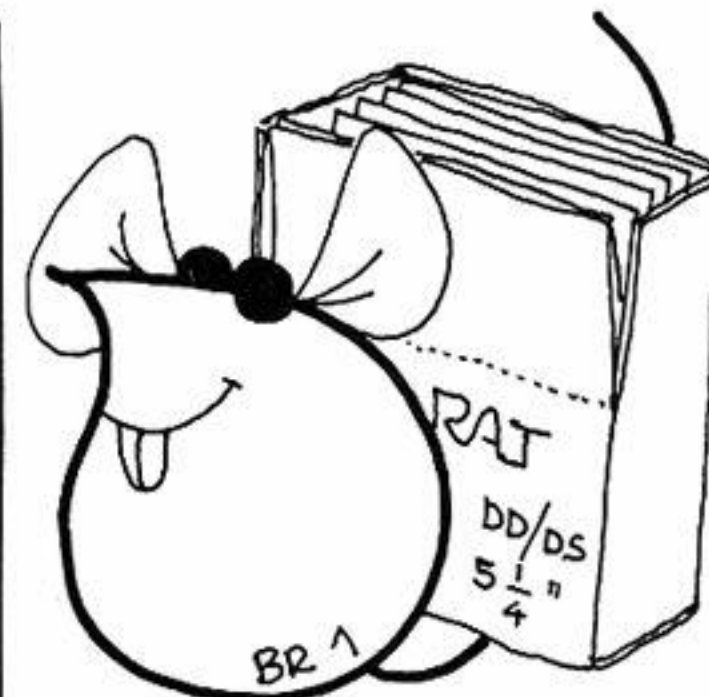
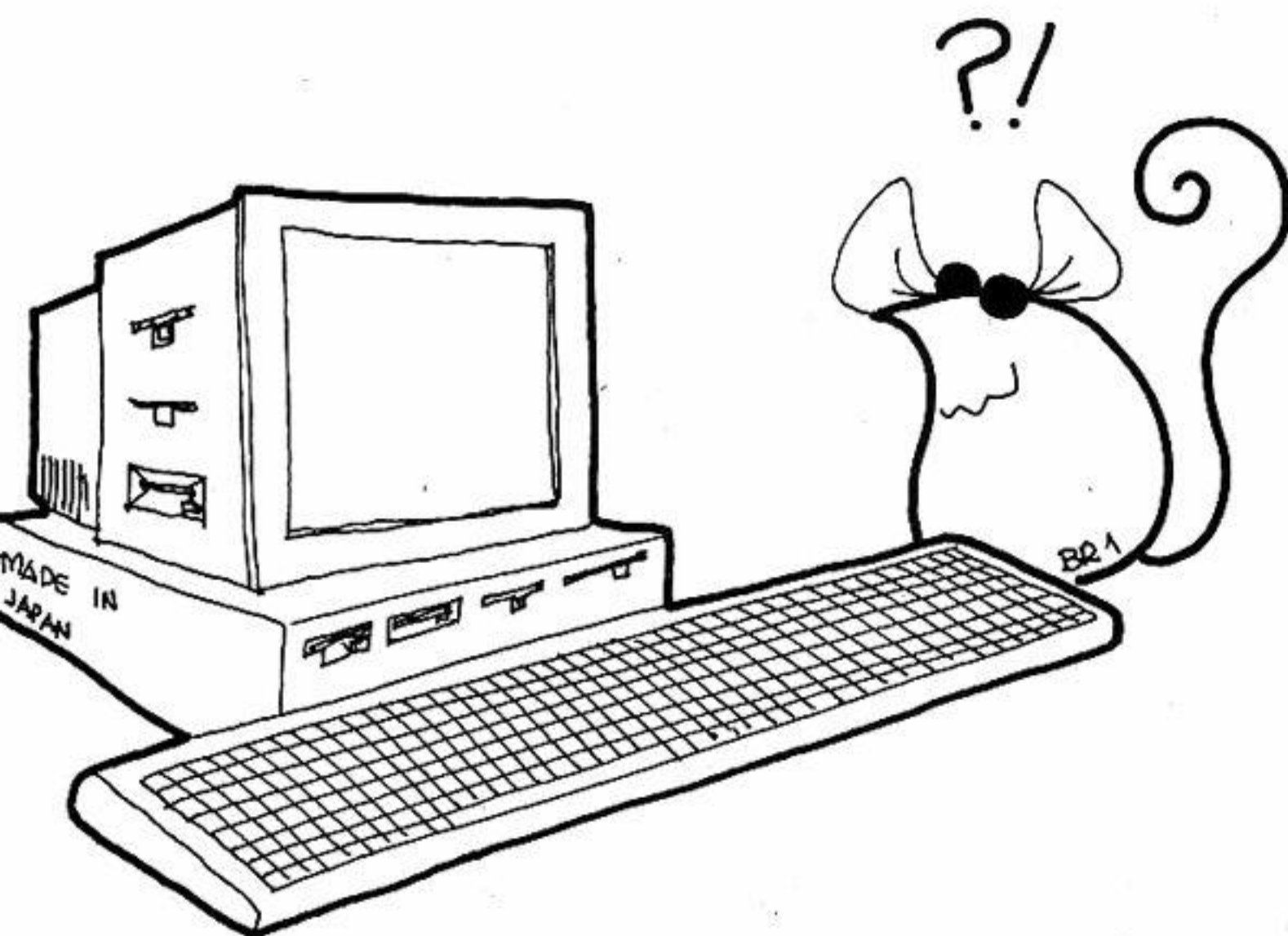
Vengono quindi effettuate altre 10 estrazioni e si confrontano, al termine, le percentuali, determinate ora, con quelle precedenti (memorizzate nel vettore P4, vedi riga 580).

Dal momento che siamo all'inizio dell'elaborazione (appena 20 estrazioni) è molto probabile che le differenze tra prima e dopo siano rilevanti; se la differenza percentuale risulta maggiore di quella digitata in fase di input (riga 140), vengono effettuate altre 10 estrazioni; e così via, finchè la differenza (lo ripetiamo, misurata in percentuale) tra ciascuna percentuale di ciascuna classe non risulterà inferiore (o pari, righe 470-550) alla variabile PP.

In pratica si tratta di esaminare una "situazione" ogni 10 estrazioni e stabilire la variazione tra quella precedente e quella successiva finchè la differenza è contenuta nei limiti imposti all'inizio.

A questo punto i dati ottenuti dovrebbero essere sufficientemente attendibili e possono rispecchiare, per grandi linee, la situazione generale che dovrebbe, grosso modo, venir riprodotta identica ai successivi passaggi.

Viene infatti introdotta una nuova elaborazione ma stavolta il numero di estrazioni non sarà 10, ma superiore, e determinato dall'incremento della variabile "T" (riga 260 e 550). Dopo un congruo numero di successive estrazioni (il cui valore viene sempre riportato in cima alla colonna che le riguarda) e corrispondenti controlli sulle variazioni percentuali, si ottengono cinque colonne, ognuna delle quali dovrebbero rappresentare successive ap-

```
100 REM CERTO, CERTISSIMO, ANZI: PROBABILE
110 REM DIVAGAZIONI SULLE "PROIEZIONI" STATISTICHE
120 :
130 REM INPUT DATI
140 PRINT CHR$(147):INPUT"PERCENTUALE";PP
150 LL=RND(-1):    REM ATTENZIONE!!!!
160 INPUT "N. ELEMENTI";NE: B(1)=1E38
170 FOR I=1 TO NE
180 PRINT"RANGE MAX. ELEM. N."I;:INPUT B(I)
190 IF B(I) < B(I-1) THEN PRINT"ERRORE": GOTO 180
200 NEXT
210 INPUT"PRESSIONE TASTO(S/N)";PT$: PT=0: IF PT$="S" THEN PT=1
220 :
230 REM INIZIO ELABORAZIONE
240 PRINT CHR$(147): NC=1: K=1: T=1
250 FOR I=1 TO NE: P3(I)=100: NEXT
260 FOR W=1 TO 10*T
270 X=B(NE)  : GOSUB 420
280 G=1:FOR E=1 TO NE
290 IF Y3 > B(E-1) THEN G=E
300 NEXT E: A(G)=A(G)+1
310 NS=0:FOR I=1 TO NE: NS=NS+A(I):NEXT
320 REM VISUALIZZA DATI
330 PRINT CHR$(19)"%=";PP;TAB(K*7)NS:PRINT
340 FOR I=1 TO NE: PRINT A(I)
350 NEXT I: NEXT W: NC=NC+1
360 PRINT CHR$(19): PRINT
370 REM DETERMINAZIONE PRESSIONE TASTO PER PROSEGUIRE
380 GOSUB 430: IF PT=1 THEN GOSUB 610
390 GOTO 260:REM RIPRENDE CICLO
400 :
410 REM GENERAZIONE NUMERO CASUALE
420 Y1=RND(1)*X: Y2=INT(Y1):Y3=Y2+1: RETURN
430 SM=0: FOR J=1 TO NE: SM=SM+A(J): NEXT
440 REM DETERMINAZIONE PERCENTUALE ELABORATA
450 FOR J=1TO NE: P2(J)=INT((A(J)*100/SM)*100)/100: NEXT
460 FOR J=1 TO NE: PRINT TAB(7*K)P2(J)"▮     ": NEXT
470 H=0: PRINT: PRINT
480 :
490 REM ESAMINA SE VALORI PRECEDENTI < ATTUALI
500 FOR J=1 TO NE
510 IF P1(J)<>0 THEN P3(J)=INT(100*((P2(J)-P1(J))/P1(J))*100)/100
520 P4$=LEFT$(STR$(P4(J))+"      ",5)
530 PRINT P4$ TAB(K*7) P3(J)"▮    "
540 IF ABS(P3(J)) < PP THEN H=H+1
550 NEXT J: IF H> NE-1 THEN K=K+1:T=T+1:IF K=5 THEN SYS49152,0:END
560 :
570 REM MEMORIZZA VALORI ATTUALI IN P4()
580 FOR J=1TO NE: P1(J)=P2(J):P4(J)=P2(J): NEXT: RETURN
590 :
600 REM ATTESA PRESSIONE TASTO
610 GET A$: IF A$="" THEN 610
620 RETURN
```

Un problema relativo ad una corretta visualizzazione è tipico dei programmi di statistica

prossimazioni verso il valore ideale. Ma non sempre è così.

UN CASO PRATICO

Per usare il programma, e per semplificare le cose, ci riferiremo alle tabelle di queste pagine e all'estrazione di numeri compresi nell'intervallo prima accennato (1/1000; 1001/2000).

Dopo aver fatto partire il programma (Run) rispondete con 10 alla domanda relativa alla percentuale.

Per ciò che riguarda il "N. di elementi" digitate 2.

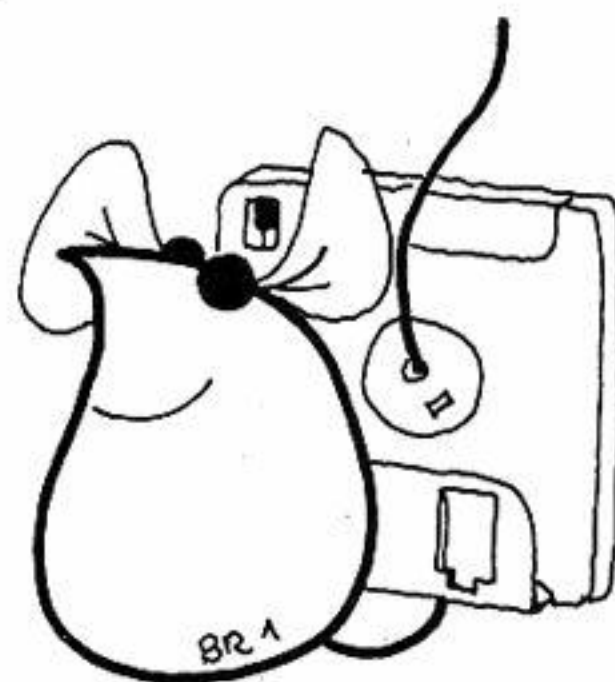
Poichè abbiamo deciso le estrazioni dei numeri tra 1 e 2000, alla due successive domande ("Range max. elem. n. 1 e n.2") rispondete, rispettivamente, con 1000 e 2000. Rispondete, infine, con "S" (e Return) alla domanda sulla pressione del tasto; ciò vi consentirà di esaminare con calma, una per una, le varie elaborazioni effettuate.

A questo punto lo schermo verrà cancellato e comparirà, in alto a sinistra, la percentuale dapprima indicata (10%) e verranno compiute alcune elaborazioni.

La prima colonna a sinistra rappresenta la quantità di elementi estratti appartenenti (in alto) all'intervallo 1-1000 e (in basso) a 1001-2000. Al fianco destro di 10% compare il valore 10, che indica il numero di estrazioni effettuate. Le due colonne in basso per il momento non rappresentano alcun dato utile e sono prive di significato.

Premendo un tasto qualsiasi l'elaborazione riprende; alla fine noteremo che sono stati estratti, dall'inizio, 20 valori di cui alcuni appartenenti al primo intervallo (prima colonna in alto, prima riga) ed altri al secondo (prima colonna in alto seconda riga). Le due colonne sottostanti ora assumono un ben preciso significato: la prima non è altro che la riproduzione fedele di ciò che era riportato (nella prima elaborazione) nella seconda colonna in alto. La seconda colonna, invece, indica lo scostamento percentuale rilevato tra la precedente elaborazione e quella "attuale".

CHIARIAMO

Per meglio chiarire le cose, riferiamoci alla tabella di figura 4, scaturita dal nostro C/64; si sottolinea che, con il vostro elaboratore, i valori possono esser diversi.

La tabella di figura 4, come tutte le altre, indica ciò che compare sul video giunti alla fine dell'elaborazione.

La sua interpretazione è la seguente:

Dopo 50 estrazioni il 52% dei numeri estratti apparteneva al primo range (1-1000) ed il 48 al secondo (1001-2000). Solo a questo punto (cioè do-

po 50 estrazioni) la differenza riscontrata tra il precedente gruppo di 10 estrazioni (cioè da 31 a 40) e quello "attuale" (cioè da 41 a 50) è risultata inferiore al limite massimo imposto all'inizio (cioè 10%); risulta infatti che il primo gruppo segnalava un differenza (in più) del 3.99%, mentre il secondo gruppo registrava un differenza (in meno) del 4%

Passando alla seconda colonna le statistiche vengono effettuate non più ogni 10 estrazioni, ma ogni 20. Al termine delle 70 estrazioni risulta che la percentuale è del 54.28 (primo gruppo) e del 45.71 (secondo gruppo): ciò significa che, aumentando le estrazioni, le percentuali, lungi dall'approssimarsi al 50% (valore teorico) divergono ulteriormente!

Si ritorna ad una più o meno accentuata normalità dopo 140 estrazioni (di cui, vedi prima colonna, 68 al primo gruppo e 72 al secondo).

Impostando una percentuale minore si nota che non sempre tale metodo permette approssimazioni migliori.

Si noti, infatti, la tabella di figura 1, relativa allo 0.5%: la divergenza è praticamente eguale alla tabella del 10%, ma ciò si verifica dopo ben 470 estrazioni.

Una minore precisione, dovuta ad una percentuale tollerabile del 20% di figura 5 (nettamente superiore alle prime due) indica un risultato praticamente identico dopo "appena" 110 estrazioni.

La percentuale di 1% (figura 2) porta, dopo 430 estrazioni, a risultati meno attendibili della figura 3 (5%) che, dopo sole 200 estrazioni, determina il risultato migliore delle cinque impostazioni: 50.5 e 49.5 per cento.

Sorge spontanea una domanda: esistono altre procedure grazie alle quali giungere al fatidico 50% in entrambi i casi?

SFIDA CONCLUSIVA

Il viaggio nella statistica è uno dei più affascinanti che un utente di home computer può compiere.

Gli strumenti a disposizione sono tra i più semplici possibili: qualche vettore, un uso accorto della funzione RND, un uso intensivo delle percentuali, di tecniche di arrotondamento, di soluzione di problemi che a mano a mano si presentano.

L'idea l'abbiamo data: tocca a voi, ora, elaborare un programma in cui si possa determinare l'algoritmo più idoneo in modo da pervenire al numero ottimale di estrazioni in base ad una situazione stabilita a priori.

EQUAZIONI DIFFERENZIALI DI ORDINE PRIMO E SECONDO

Come rendere utili, anche a scuola, i piccoli computer della Commodore

di **Simone Trevisan**

Due brevissimi listati vi aiutano in matematica

Senza pretendere di dare una definizione rigorosa di "equazione differenziale", possiamo dire che essa è un'equazione in cui possono comparire, oltre alle normali operazioni (somma algebrica, radice, logaritmo e così via) anche quelle d'integrazione e di derivazione.

Tali elaborazioni vengono spesso compiute nell'analisi di sistemi fisici, come quelli meccanici, termici, elettrici. Solitamente, per evitare i laboriosi calcoli introdotti dalla risoluzione delle equazioni differenziali, si preferisce operare sulla "trasformata" di Laplace.

Grazie al computer, però, i calcoli più complessi si traducono in un'attesa di pochi secondi.

Il primo programma presentato in queste pagine risolve le equazioni differenziali di primo ordine. L'altro listato, oltre a risolvere le equazioni di primo ordine (ponendo il coefficiente di y'' uguale a zero) provvede ad elaborare i valori di equazioni del secondo ordine.

```
INTERVALLO A/B 0  1
Y( 0 )= 1
PASSO= .1
Y( .1 )= 3.80315616
Y( .2 )= 4.21882518
Y( .3 )= 4.68072754
Y( .4 )= 5.21856701
Y( .5 )= 5.81464722
Y( .6 )= 6.48393476
Y( .7 )= 7.23412892
Y( .8 )= 8.0737387
Y( .9 )= 9.01216803
Y( 1 )= 10.0598098

100 DEF FN F(X)=X+Y
```

```
INTERVALLO A/B:  0  1
Y( 0 )= 1
PASSO=  .05
Y( .05 )= 1.04880886
Y( .1 )= 1.09544514
Y( .15 )= 1.14017546
Y( .2 )= 1.183216
Y( .25 )= 1.22474493
Y( .3 )= 1.26491113
Y( .35 )= 1.30384056
Y( .4 )= 1.34164088
Y( .45 )= 1.37840498
Y( .5 )= 1.41421369
Y( .55 )= 1.44913781
Y( .6 )= 1.48323985
Y( .65 )= 1.51657526
Y( .7 )= 1.54919353
Y( .75 )= 1.58113904
Y( .8 )= 1.61245178
Y( .850000001 )= 1.64316793
Y( .900000001 )= 1.67332034
Y( .950000001 )= 1.70293895
Y( 1 )= 1.73205115

100 DEF FN F(X)=Y-2*X/Y
```

DATEVI DA FARE!

Già in precedenza sono apparsi, sulle pagine della nostra rivista, programmi per la visualizzazione di funzioni matematiche.

I programmi proposti dal lettore Trevisan rappresentano un ulteriore contributo allo studio della matematica con l'ausilio del computer.

Peccato che le applicazioni "pratiche" siano state solo accennate e, per di più, senza indicazioni precise sulle relazioni matematiche e sulle procedure da usare per implementarle sul computer.

La nostra proposta è quindi scontata: provate a suggerire casi "concreti" di funzioni relative a situazioni reali, elaborabili con i listati pubblicati (o modificati per l'occasione).

E, soprattutto, illustrate il significato dei dati elaborati.

A che possono servire gli approfondimenti suggeriti? Semplice: simulazioni con il calcolatore. Che cosa accaderebbe, ad esempio, se...

I più bravi possono inviare i risultati delle loro ricerche (SU DISCO!) corredati da articoli (è di rigore l'uso di un qualsiasi W/p) che descrivano accuratamente sia la funzione studiata sia il significato delle elaborazioni eseguite.

Systems Editoriale
Viale Famagosta 75
20142 MILANO
tel. 02/84.67.348

IL PRIMO PROGRAMMA

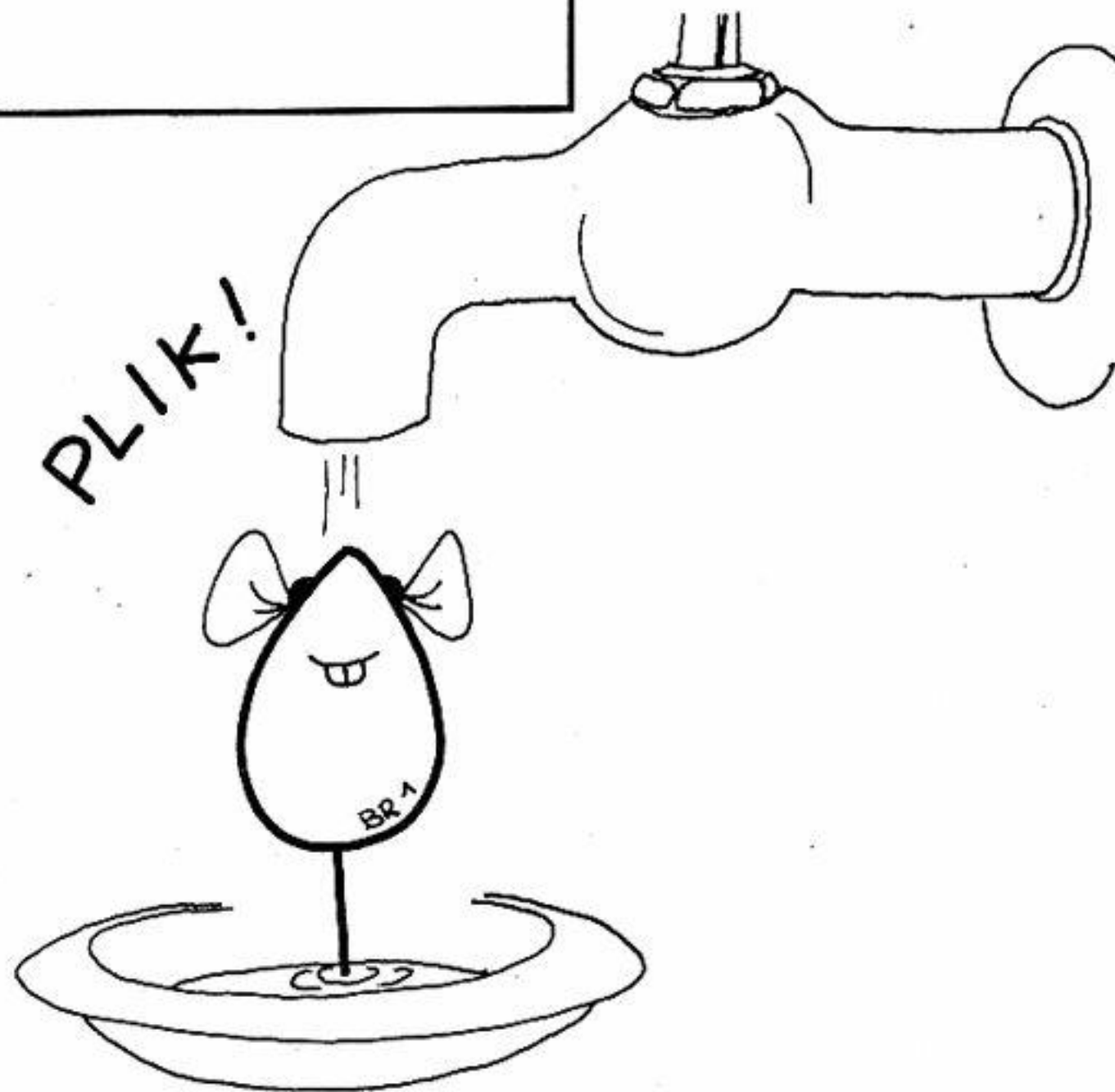
L'algoritmo usato è quello di Runge - Kutta, che risulta molto più preciso di quello di Eulero.

L'equazione, che i libri di testo riportano tutti nella forma...

$y' = f(x, y)$

...deve essere "computerizzata" nella forma: Def Fn().

Si osservi, a tal proposito, il modo in cui viene introdotta e memorizzata la funzione in oggetto. Con il C/128 viene ridefinito il tasto F1. Chi dispone, invece, del solo C/64 deve trascrivere le righe da 65 a 100 (indicate a parte), invece del gruppo di righe 65 / 100 presenti nel listato idoneo per il solo C/128.

La precisione della risoluzione dipende dal valore della variabile H (passo): tanto più questo sarà piccolo, tanto più preciso sarà il risultato. L'attesa, come intuitivo, risulterà proporzionalmente più lunga.

```
5 REM  PROGRAMMA BY TREVISAN SIMONE
6 PRINT CHR$(147)"VERSIONE C/128 (MODO 128)"
10 PRINT" EQUAZIONI DIFFERENZIALI I ORDINE METODO RUNGE-KUTTA"
20 PRINT"ESEMPI DI EQUAZIONE:".PRINT:PRINT
25 PRINT"Y'=X+Y        INTERVALLO 0;1   H=.1  Y(0)=1
30 PRINT"Y'=Y-2*X/Y  INTERVALLO 0;2   H=.05 Y(0)=1
40 REM ***INSERIMENTO EQUAZIONE NELLA FORMA Y'=F(X,Y)***
60 PRINT:INPUT "Y'= ";F$
65 :
70 KEY1,"100 DEF FN F(X)="+F$+CHR$(13)+"RUN 90"+CHR$(13)
80 PRINT:PRINT"PER FAVORE PREMI F1":END
90 REM NON CANCELLARE QUESTA LINEA!!! (VEDI RIGA 70)0
95 :
100 DEF FN F(X)=Y-2*X/Y
105 REM ***INSERIMENTO INTERVALLO,CONDIZIONI INIZIALI,PASSO***
107 PN=3:INPUT"VIDEO/STAMPANTE (V/S)";PN$:IFPN$="S"THEN PN=4
110 CLOSE1:OPEN1,PN: INPUT"INTERVALLO A - B";A,B
115 IF A>=B THEN 110
117 PRINT#1,"INTERVALLO A/B: ";A;B
120 PRINT"Y(";A;")=";:INPUT Y:PRINT#1,"Y("A")="Y
130 H=.1:INPUT"PASSO (PER DEFAULT H=.1) ";H
132 PRINT#1,"PASSO= ";H
135 REM ***RISOLUZIONE DELL'EQUAZIONE CON ALGORITMO DI RUNGE KUTTA***
140 FOR I=A  TO B-H/2 STEP H
150 K1=H*FN F(I):Y=Y+K1/2
160 K2=H*FN F(I+H/2):Y=Y+K2/2-K1/2
170 K3=H*FN F(I+H/2):Y=Y-K2/2+K3
180 K4=H*FN F(I+H):Y=Y-K3
190 AY=1/6*(K1+2*K2+2*K3+K4)
200 Y=Y+AY:PRINT#1,"Y(";I+H;")=";Y
210 NEXT:CMD1: LIST 100: PRINT#1:CLOSE1
```

MODIFICA LISTATO 1 PER C/64

```
65 PRINT"POSIZIONATI SULLA RIGA 100, E PREMI RETURN DUE VOLTE"
70 PRINT:PRINT
80 PRINT"100 DEF FN F(X)=";F$:PRINT"RUN 100":PRINT:PRINT.END
95 :
100 DEF FN F(X)=Y-2*X/Y
```

IL SECONDO PROGRAMMA

Tale programma, valido per il solo C/128, risolve le equazioni differenziali di primo e secondo ordine.

Da tastiera vengono chiesi alcuni dati:

- la funzione f(x), ossia quel termine in cui non compare la variabile y (o le sue derivate);
- il passo H;
- l'intervallo di risoluzione;
- i tre coefficienti, rispettivamente di y'', y' e y.

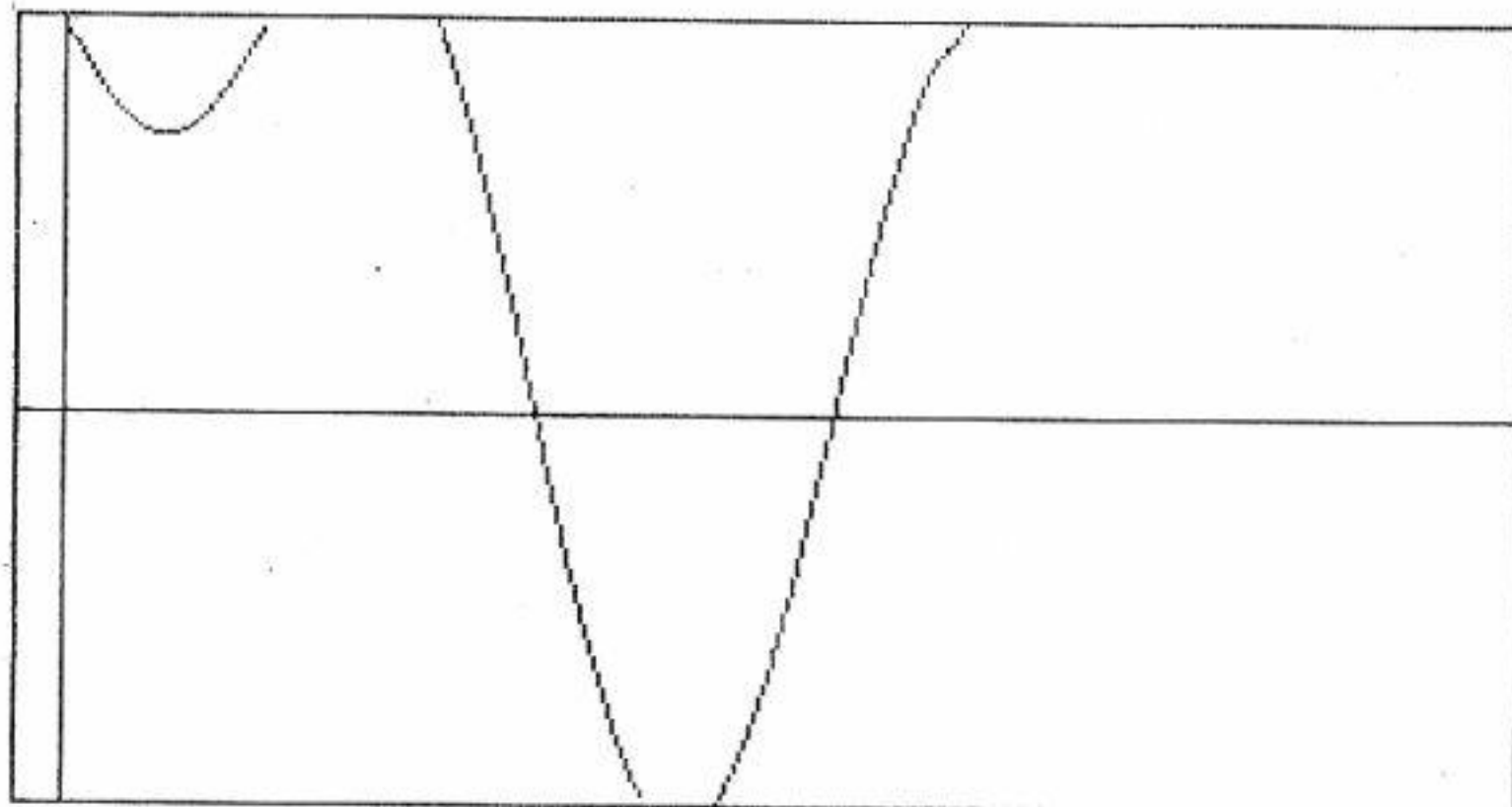
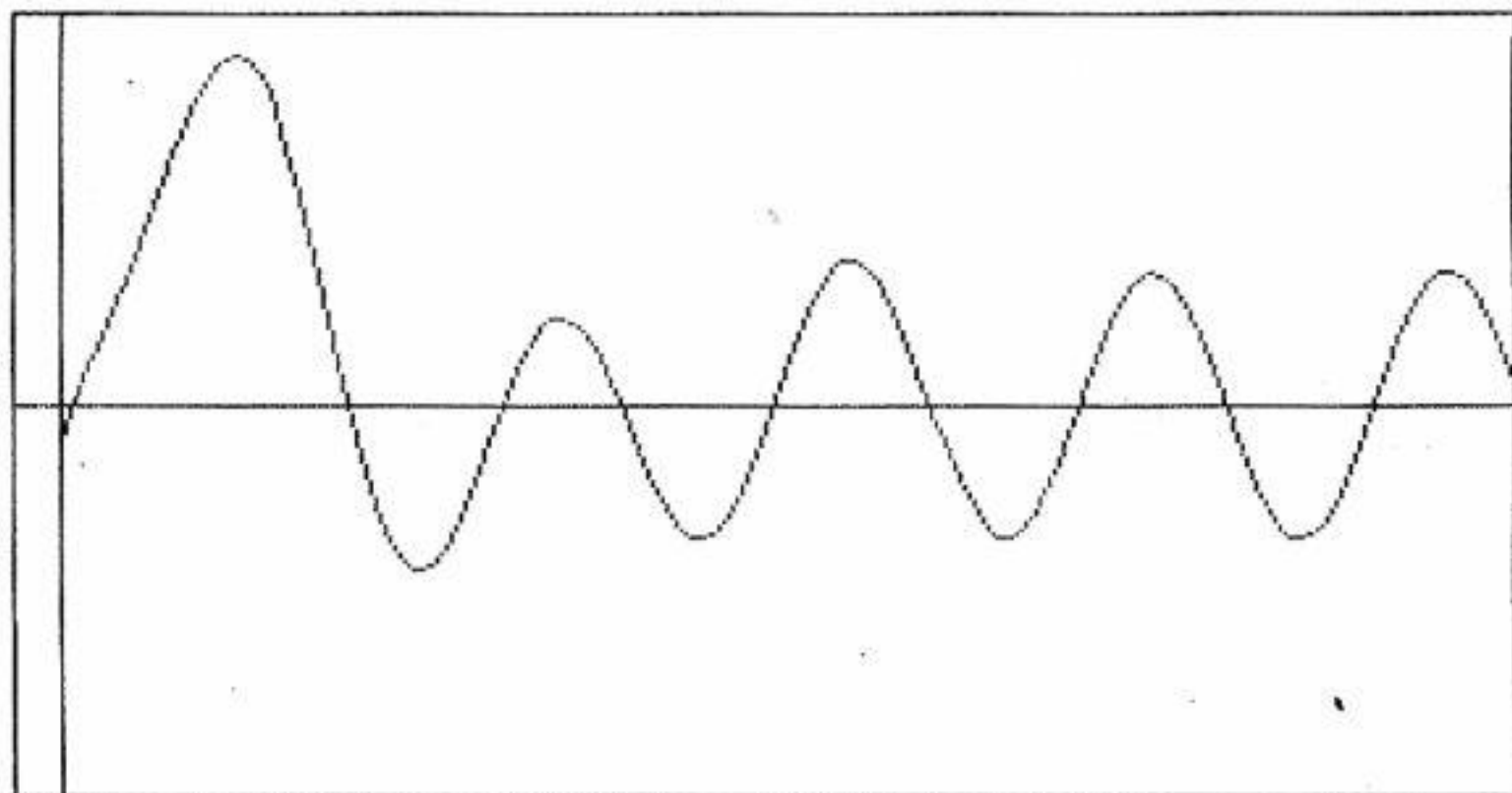
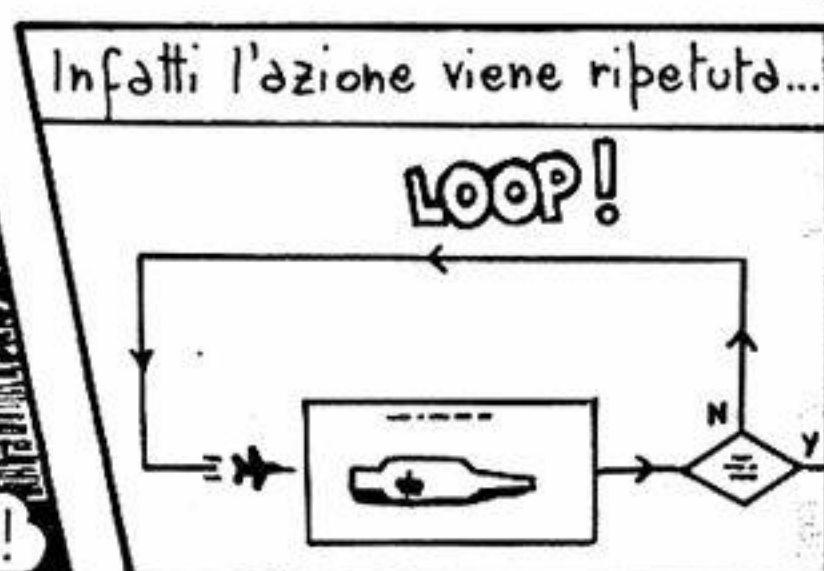
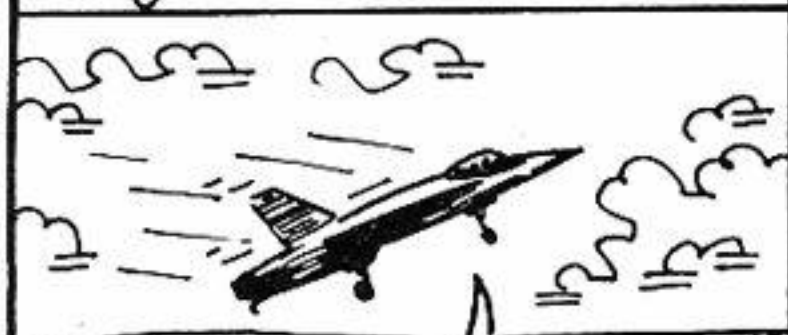
Si noti che il programma funziona egregiamente per valori dei coefficienti non minori a 0.001 poichè, per tali valori, vengono introdotti errori talmente grandi da non rendere più significativi i risultati stessi.

E' piuttosto suggestivo visualizzare, graficamente, i risultati dell'elaborazione poichè rappresenta la risposta di un sistema fisico ad un eventuale ingresso che coincida con la funzione f(x) introdotta.

Si consideri, ad esempio, un forno e si immagini di accenderlo posizionando la manopola del gas per una temperatura di 80 C.

La temperatura non raggiungerà subito la temperatura impostata, ma solo dopo un certo tempo, strettamente dipendente dalle caratteristiche del forno stesso (dimensioni, capacità termica, eventuali dispersioni e così via).

L'andamento della temperatura può essere ottenuto risolvendo, appunto, un'equazione differenziale del primo ordine in cui l'ingresso, in questo caso, è un "gradino" (di potenza).

Con i programmi di queste pagine si può simulare la risposta a gradino, a rampa, ad ingresso sinusoidale di un sistema fisico, quale ad esempio una rete elettrica.

```
120 DEF FN F(X)=100*SIN(X)
```

```
120 DEF FN F(X)=15 * SIN(X) * 15 * COS(X)
```

```
10 REM PROGRAMMA BY TREVISAN SIMONE
20 REM VERSIONE 128 IN MODO 128
25 PRINT"EQUAZIONI DIFFERENZIALI 2 ORDINE METODO EULERO (PER SOSTITUZIONE)"
30 PRINT"ESEMPI:":PRINT
32 PRINT"F(X)=100  H=0.1  INTERVALLO 0,10  COEFF.A =1 B=1 C=1
33 PRINT"CONDIZIONI INIZ Y(0)=0  Y'(0)=0":PRINT
34 PRINT"F(X)=100*SIN(X) H=0.1 INTERV. 0,10 COEFF.A =.8 B=1 A=.5
35 PRINT"CONDIZIONI INZ Y(0)=0  Y'(0)=0
40 REM ***INTRODUZIONE TERMINE NOTO***
60 PRINT"AY''+BY'+CY=F(X)[illegible]"
70 INPUT "F(X)= ";F$
80 KEY 1,"120 DEF FN F(X)="+F$+CHR$(13)+"RUN 110"+CHR$(13)
90 PRINT "PREMI F1":END
100 REM ***INTRODUZIONE INTERVALLO,CONDIZIONI INIZIALI,PASSO,COEFFICIENTI***
110 INPUT"INTERVALLO ";X1,X2
115 IF X1>=X2 THEN 110
120 DEF FN F(X)=100*COS(X)+SIN(X)↑2
130 PRINT"Y(";X1;")= ";:INPUT Y
140 PRINT"Y'(";X1;")= ";:INPUT Z
150 H=.1:INPUT "PASSO (PER DEFAULT H=.1)";H
160 INPUT"COEFFICIENTE DI Y'' ";A
170 INPUT "COEFFICIENTE DI Y' ";B
180 INPUT "COEFFICIENTE DI Y  ";C
190 REM ***RISOLUZIONE EQUAZIONE DIFFERENZIALE***
200 FOR X=X1 TO X2+H STEP.H
210 PRINT"Y(";X;")= ";Y
220 R=Z:T=Y
230 Y=Y+H*R
240 Z=Z+H*(FN F(X)-B*Z-C*T)/A
250 NEXT:PRINT
260 PRINT CHR$(18) "PER ESAMINARE IL GRAFICO PREMI UN TASTO"
270 GET A$:IFA$="" THEN 270
1000 REM ***VISUALIZZAZIONE DELLA RISOLUZIONE DELL' EQUAZIONE DIFFERENZIALE***
1010 GRAPHIC 1,1:DRAW 1,0,0 TO 319,0 TO 319,199 TO 0,199 TO 0,0
1020 DRAW1,10,0 TO 10 ,200
1030 DRAW1,0,100 TO 320,100
1040 X3=10:Y3=100
1050 FOR X=X1 TO X2+H STEP H
1060 IF 100>=ABS(-Y) AND X3>=0 AND Y3>=0 THEN DRAW1,X3,Y3 TO X*20+10,100-Y
1070 X3=X*20+10:Y3=100-Y
1080 R=Z:T=Y
1090 Y=Y+H*R
1100 Z=Z+H*(FN F(X)-B*Z-C*T)/A
1110 NEXT:END
```

UNA SCHERMATA TUTTA DA TRADURRE

Come utilizzare le immagini hi-res elaborate con i più diffusi tool grafici in commercio

di **Domenico Pavone**

Un solo programma è sufficiente per caricare schermate hires

Il tema affrontato in queste pagine porta, inevitabilmente, a puntualizzare temi di interesse generale: grafica hi-res, Assembly, e gestione dell'immancabile MacroAssembler.

Anche i più "materiali", comunque, non avranno di che lamentarsi: teoria a parte, si ritroveranno in ogni caso con un programma di conversione di file grafici dallo standard Koala ad un formato che li renda direttamente caricabili anche da Basic.

UN PO' DI GEOGRAFIA

Analizziamo, dunque, il contenuto dei file generati da tre diversi programmi di vasta diffusione: Koala, Doodle e Print Shop.

Questi rappresentano, come è possibile notare dalla figura 1, tre filosofie diverse, capite le quali diventa poi facile interpretare (e manipolare) qualunque altro sistema di memorizzazione grafica.

Una prima differenza fondamentale riguarda il tipo di hi-res utilizzata, che è quella multicolor per il Koala, mentre Doodle e Print Shop sfruttano il sistema bitmap standard.

Il che, tradotto in parole povere, significa risoluzione di 160 x 200 nel primo caso, con possibilità di accostare fino a 4 colori nell'ambito di una "cellula" di pixel (8x8), e di 320 x 200 nel secondo caso, quindi con una maggiore definizione, ma minori possibilità cromatiche (solo 2 colori in ogni griglia 8x8).

Il C/64, per visualizzare una schermata in hi-res, necessita in particolare dei seguenti elementi:

- Una bitmap, ovvero un'area di memoria contenente lo stato (acceso / spento) di ogni pixel. Nel modo multicolor, ad ogni coordinata orizzontale corrisponderanno 2 bit. In entrambi i casi, quindi, la bitmap sarà composta da 64000 bit, e poichè ogni cella di memoria contiene 8 bit (= 1 byte), avremo un totale di 8000 locazioni (64000 / 8). Utilizzando i primi 16 Kbyte (suo limite fisico di gestione) per il VIC, si è soliti collocare quest'area a partire da 8192 ($2000).
- Una mappa colore, ovvero un'altra area di 1000 locazioni, ognuna delle quali influenza il colore di una griglia di pixel 8x8. Quest'area, quando si entra in hi-res, non è altro che la memoria di schermo, quella in cui (in modo testo) è possibile pokarvi i caratteri da visualizzare (1024 - 2023).

  In effetti, purchè all'interno dei 16 Kbyte del VIC, sarebbe possibile rilocarla altrove, ma limitiamoci alle condizioni di default del sistema; in caso contrario l'argomento si espanderebbe a dismisura.
- Nel caso del multicolor, è poi necessaria una seconda mappa colore, rappresentata dalle consuete locazioni da 55296 a 56295, più il colore di fondo (la solita 53281).

I file ottenuti dai tool grafici, quindi, devono contenere tutti, o quasi tutti, i dati in questione.

Ed in effetti, tornando alla figura 1, li si può rintracciare agevolmente, anche se organizzati in diversa maniera e a diversi indirizzi.

Se, infatti, caricassimo in memoria un file Koala (multicolor) con Load....8,1 avremmo le varie componenti della schermata agli indirizzi specificati in figura.

SCHERMATE PERSONALI

Nella enorme quantità di software per C/64, alcune scelte risultano pressochè costanti in tutta la fascia di utenza del più popolare computer della Commodore.

Tra queste, i programmi finalizzati alla grafica in alta risoluzione giocano certamente la parte del leone, secondi, forse(?), solo alla miriade di copiatori e sprotettori tritatutto del mercato parallelo.

Grazie a questi tool, viene aggirata la intrinseca difficoltà, con il 64, di creare immagini degne di nota sfruttando l'hi-res.

Disegnare da programma la più semplice figura geometrica, a causa della nota mancanza di comandi Basic adatti allo scopo, diventa infatti estremamente complicato e, comunque, abbastanza noioso anche per un esperto programmatore in linguaggio macchina.

E' molto più pratico (e divertente) affidarsi alla più familiare leva del joystick, sfruttando al meglio le icone del Koala, o le opzioni "tastierecce" del Doodle, per non parlare dell'ancora più semplice "Print Shop" e similari.

Il risultato finale, salvo diversa scelta dell'utente, è quasi sempre un nuovo file su disco, contenente lo sfavillante frutto delle nostre fatiche: una bella schermata, possibilmente ben colorata, pronta per essere mostrata con orgoglio ad amici e parentado assortito.

La cosa non presenta alcuna difficoltà, a patto di avere la pazienza di caricare il particolare tool con il quale si è elaborata l'immagine, attendere che venga caricato il file grafico, e così via.

L'uso fine a se stesso delle schermate, peraltro, non è certo il massimo cui si possa aspirare.

Com'è facile intuire, sarebbe ben più stimolante inserire i disegni in questione all'interno di propri programmi, rendendoli indipendenti dal tool che li ha generati e quindi visualizzabili quando e come ci pare.

La velocità di caricamento, ovviamente, è condizionata dal drive usato

Un file Doodle, che non utilizza il multicolor, userà solo una mappa colore (oltre la bitmap), mentre il Print Shop, addirittura, contiene unicamente la bitmap.

MANOVRE PRELIMINARI

Una volta padroni degli indirizzi in cui sono memorizzati i vari elementi dell'immagine hi-res, tutto diventa più facile... o quasi.

In pratica, per visualizzare direttamente una schermata, occorre prima caricare il file, dopodichè trasferire al posto giusto le varie aree di memoria ed attivare l'alta risoluzione.

Il problema, in effetti, non si pone per il Print Shop, come già trattato sul n.41 della rivista (pagina 88), in quanto è sufficiente alterare i puntatori di inizio del Basic facendoli "mirare" ad 8192, effettuare un normale caricamento e riempire con un colore a scelta l'area di schermo.

Per Koala e Doodle, invece, qualche trasferimento si rende obbligatorio.

Vediamo di approfondire e concretizzare l'argomento, limitandoci d'ora in avanti ai file di tipo Koala: sarà poi facile applicare quando appreso anche ai file Doodle, strutturalmente più semplici, aiutandosi con gli indirizzi specificati in figura 1.

Supponiamo, dunque, di avere elaborato e salvato su disco una schermata con il Koala (o Koala-Painter che dir si voglia).

La prima operazione da compiere non riguarda direttamente la grafica, ma piuttosto il Dos.

Si tratta, come forse avrete capito, di cambiare nome al file generato da Koala, che adopera un fastidioso e poco maneggevole simbolo di "picche" in campo inverso come primo carattere (codice Ascii 129).

I più esperti si sbizzarriscano con le "diavolerie" che preferiscono, gli altri possono adoperare questa semplice tecnica:

• Caricare la directory del disco contenente il file, quindi impartire List e usare il tasto Stop per lasciare sul video il nome che si intende rinominare.

• Digitare New (e Return), quindi portarsi col cursore sulla riga in questione, eliminare la scritta PRG e premere di nuovo Return. In pratica, si è creata una riga Basic di numero 40 (i blocchi del file).

• Modificare la riga aggiungendo "A$=" (senza le virgolette) prima del nome del file, quindi inserire quest'altra linea:

```
50 Open15, 8, 15, "r:nuovonome=" + a$
60 close 15
```

• Un bel Run, e tutto è sistemato.

MACRO ASSEMBLER: CHI CERCA, TROVA

Si è più volte ribadito che, programmando in Assembly, ordine e chiarezza sono due componenti indispensabili.

In particolare, quando si edita il contenuto di una routine, non bisogna mai dimenticare che quando riprenderemo in mano il disassemblato sorgente, magari tra qualche mese, dovrà risultare leggibile con facilità ed immediatezza: la memoria (quella nostra, non quella del C/64!) ha un limite.

Questo vuol dire abbondare con i commenti (non è, come in Basic, che occupino spazio in memoria o rallentino il programma) e, soprattutto, adoperare per le Label i nomi che richiamino immediatamente la funzione che svolgono.

La cosa vale anche per i disassemblati che trovate in queste pagine: non è detto che quanto sembra chiaro a chi li ha stilati, abbia la stessa "trasparenza" per tutti i lettori.

Un esempio, tratto da queste pagine: in riga 12 è definita l'etichetta VDMT, che identifica la matrice video, ovvero l'area in cui verrà a trovarsi la mappa colore 1 dopo il caricamento del file Koala.

Se, magari dopo avere copiato il disassemblato con l'editor, preferiste sostituire il nome della label con VIDEO, sarebbe necessario scorrere l'intero listato e modificare ogni riga dove sia presente Vdmt.

Il compito, se un elaborato è piuttosto lungo, diventa piuttosto arduo, col rischio poi di lasciarne invariata qualcuna ed accorgersene solo dopo la segnalazione di errore dell'Assembler.

Per fortuna il Macro Assembler Commodore viene ancora una volta incontro, con due potenti comandi adoperabili in modo diretto:

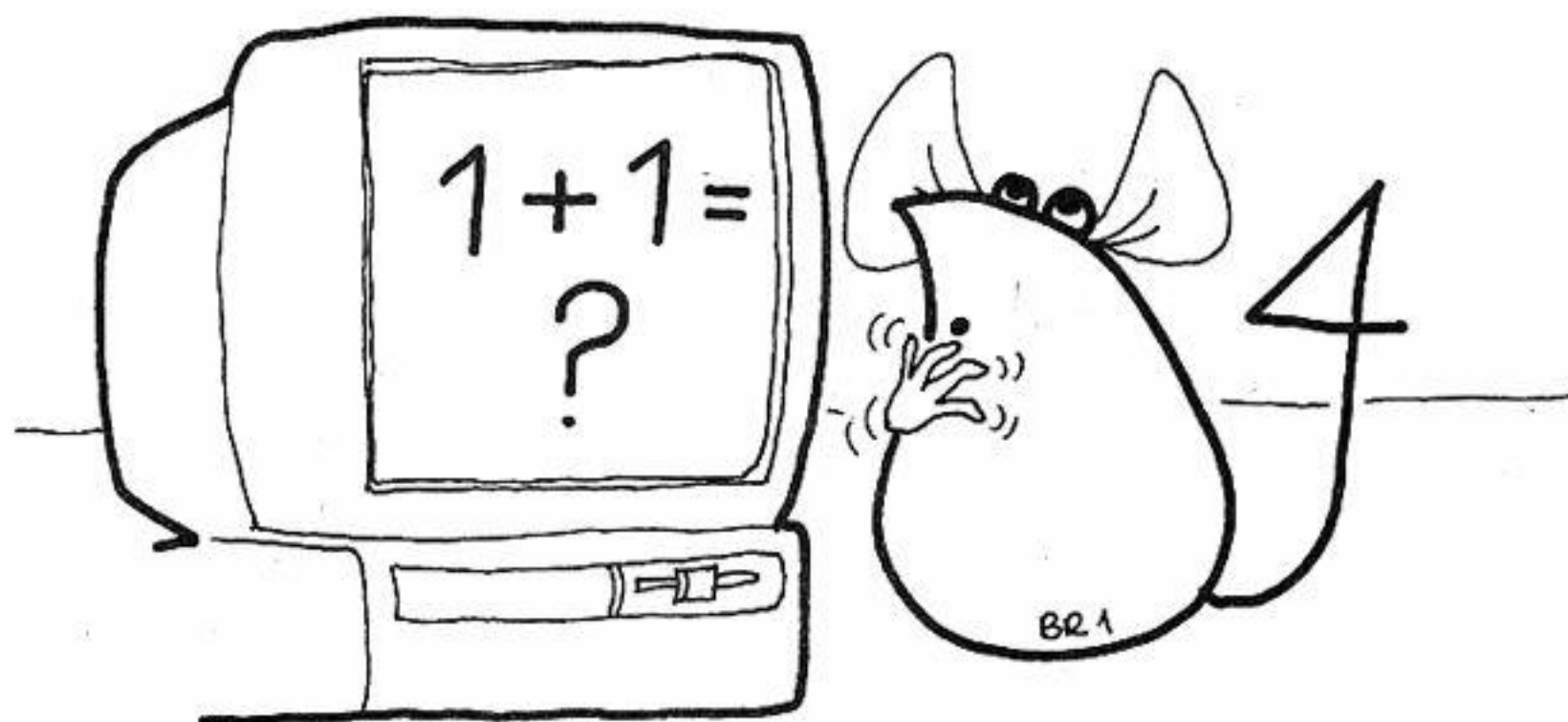
Find /stringa/, riga1 - riga2

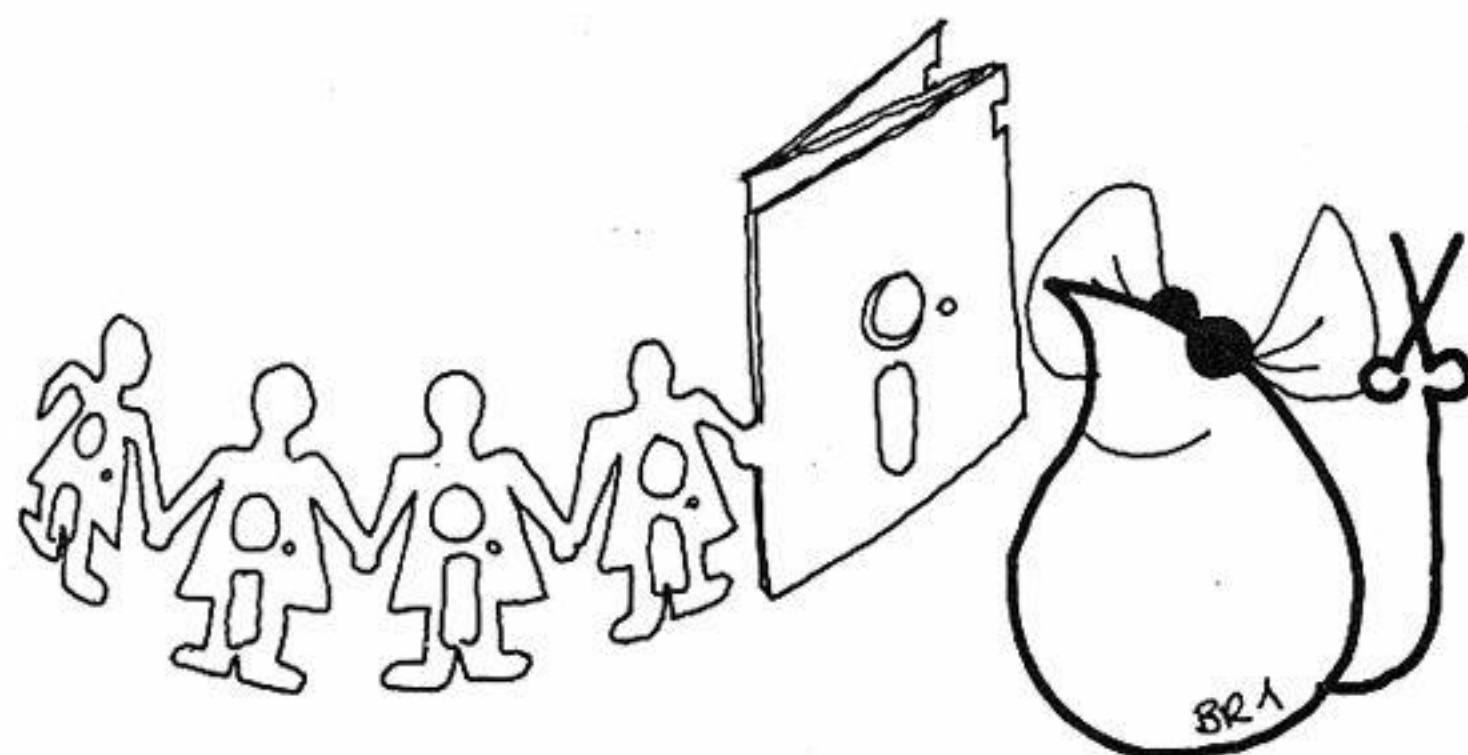
Change /stringa1/ stringa2/, riga1, riga2

Con "Find" viene rintracciata (e listata) la stringa specificata tra le due barre, mentre con Change la stringa2 sostiuirà la stringa1 in ogni sua ricorrenza, e vengono listate le righe ove ciò è avvenuto.

Dopo l'ultima barra, inoltre, è possibile specificare le linee da prendere in considerazione (la virgola è obbligatoria), seguendo le stesse (arcinote) modalità legate all'uso di List.

Se quest'ultimo parametro non è presente, la virgola non è necessaria, e l'editor scandirà tutto il disassemblato.

Un ennesimo grazie, quindi, al nostro super assemblatore.

DISASSEMBLATO "TRATTAMENTO SCHERMATE KOALA"

```
001 CHROUT  =$FFD2      dec.65490
002 INPUT   =$ABFD      dec.44029
003 SCRCLR  =$E544      dec.58692
004 BUFF    =$200       dec.512
005 SETNAM  =$FFBD      dec.65469
006 SETFIL  =$FFBA      dec.65466
007 LOAD    =$FFD5      dec.65493
008 SAVE    =$FFD8      dec.65496
009 GETIN   =$FFE4      dec.65508
010 NOMESG  =$FF90      dec.65424
011 BTMP    =$2000      dec.8192
012 VDMT    =$3F40      dec.16192
013 CLMT    =$4328      dec.17192
014 EXCBAS  =$A7AE      dec.42926
015 ILLEG   =$B248      dec.45640
016 PRT     =$FB        dec.251
017 ARR     =$FD        dec.253
018         *=$C000     Start = 49152
019 ;------------------------------------------
020         .MAC FILNAM Definisce Macro.
021         LDA #$6     Len nome file.
022         LDX #<?1    Low parametro 1.
023         LDY #>?1    Hi parametro 1.
024         JSR SETNAM  Routine Kernal.
025         .MND        Fine della Macro.
026 ;------------------------------------------
027         .MAC SALV   Definisce Macro.
028         LDA #<?1    Byte basso inizio
029         STA PRT     del Save in $FB.
030         LDA #>?1    Byte alto inizio
031         STA PRT+1   del Save in $FC.
032         LDA #PRT    Indir. puntatori.
033         LDX #<?2    Fine di Save(low)
034         LDY #>?2    Fine di Save (hi)
035         JSR SAVE    Routine Kernal.
036         LDA $90     Legge Status.
037         BNE FINE    Esce se <> 0.
038         .MND        Fine Macro.
039 ;------------------------------------------
040         JSR $E200   Prende parametro
041         STX $02     numerico dopo la
042         DEX         Sys, lo deposita
043         BEQ CONT1   in locazione 2 e
044         INX         lo controlla.
045         BEQ CONT1   Se <> 0 e <> 1,
046         JMP ILLEG   salta a Ill.Error
047 ;------------------------------------------
048 CONT1   LDX #$0     Legge caratteri
049 PRINT   LDA MSG,X   del messaggio
050         BEQ NOMFIL  fino allo zero.
051         JSR CHROUT  Routine Kernal.
052         INX         Incrementa X.
053         BNE PRINT   Salta (=JMP).
054 ;------------------------------------------
055 NOMFIL  JSR INPUT   Rout. del basic.
056         LDX #$0     Legge dal buffer
057 LEN     LDA BUFF,X  del basic fino
058         BEQ CONT2   ad uno zero.
059         INX         X=len nome file.
060         BNE LEN     Salta (=JMP).
061 ;------------------------------------------
062 CONT2   TXA         Len nome in Acc.
063         LDX #<BUFF  Low indir. nome.
064         LDY #>BUFF  Hi indir. nome.
065         JSR SETNAM  Routine Kernal.
066 ;------------------------------------------
067         LDX #$8     Periferica n.8.
068         LDY #$0     Caricam.rilocato.
069         JSR SETFIL  Routine Kernal.
070 ;------------------------------------------
071         LDA $D011   Disabilita la
072         AND #$EF    visualizzazione
073         STA $D011   dello schermo.
074 ;------------------------------------------
075         LDA #$0     0 = caricamanto.
076         TAX         Caricamento rilo-
077         LDY #$20    cato da 8192.
078         JSR LOAD    Routine Kernal.
079         BCC CONT3   Salta se tutto OK
080         JMP FINE    altrimenti esce.
081 ;------------------------------------------
```

```
082 CONT3   LDX #<VDMT   Prepara puntatori
083         LDA #>VDMT   di pagina zero
084         STX PRT      con indirizzo di
085         STA PRT+1    start per Trasf..
086         LDX #$00     e altri due con
087         LDA #$04     indirizzo di
088         STX ARR      arrivo (1024) per
089         STA ARR+1    mappa colore 1.
090         JSR TRASF    Gosub 193.
091 ;-------------------------------------
092         LDX #<CLMT
093         LDA #>CLMT
094         STX PRT      Come sopra, ma
095         STA PRT+1    per trasferire
096         LDX #$00     mappa colore 2
097         LDA #$D8     a 55296.
098         STX ARR
099         STA ARR+1
100         JSR TRASF
101 ;-------------------------------------
102         LDA $DBE8    Preleva colore
103         STA $D021    e lo deposita
104         STA $D020    in 53281 e 53280.

105 ;-------------------------------------
106         LDA $D018    Pone bitmap a
107         ORA #$8      partire dalla
108         STA $D018    locazione 8192.
109         LDA $D011    Riabilita schermo
110         ORA #$30     ed attiva l'alta
111         STA $D011    risoluzione.
112         LDA $D016    Attiva il modo
113         ORA #$10     multicolore.
114         STA $D016
115 ;-------------------------------------
116         LDA $02      Preleva parametro
117         BNE DISK     e salta se =1.
118 ATTESA  JSR GETIN    Routine Kernal
119         CMP #$0      per aspettare la
120         BEQ ATTESA   pressione di un
121         BNE FINE     tasto ed uscire.
122 ;-------------------------------------
123 DISK    LDA #00      Esclude tutti i
124         JSR NOMESG   messaggi del S.O.
125         FILNAM FILE1      Salva area
126         SALV BTMP,VDMT    bitmap.
127         FILNAM FILE2      Salva mappa
128         SALV $D800,$DBFF colore 2.
129         FILNAM FILE3      Salva mappa
130         SALV $400,$800    colore 1.

131 ;-------------------------------------
132 FINE    LDA #$C8     Esclude il modo
133         STA $D016    multicolor.
134         LDA #$1B     Riabilita modo
135         STA $D011    testo.
136         LDA #$15     Disalloca la
137         STA $D018    bitmap.
138         LDA #$0F     Colore azzurro
139         STA $D020    nel bordo.
140         LDA #$6      Colore blu
141         STA $D021    nello sfondo.
142         JSR SCRCLR   Pulisce schermo.
143         JMP EXCBAS   Torna al basic.

144 ;-------------------------------------
145 TRASF   LDX #$4
146         LDY #$0      Subroutine per
147 LOOP    LDA (PRT),Y trasferire un
148         STA (ARR),Y blocco di memoria
149         INY          di un Kbyte da
150         BNE LOOP     indir. puntato
151         INC PRT+1    da PRT a quello
152         INC ARR+1    puntato da ARR.
153         DEX
154         BNE LOOP
155 RETURN  RTS

156 ;-------------------------------------
157 MSG     .BYT 'NOME '  Messaggio per
158         .BYT 'FILE',0 Input (0=Fine).
158 FILE1   .BYT 'BITMAP' Nomi dei File
159 FILE2   .BYT 'COLOR2' da usare per
160 FILE3   .BYT 'COLOR1' i tre Save.
161         .END          Fine.
```

Anche il C/128 può caricare schermate provenienti da "Doodle" e "Print shop"

SULLO SCHERMO

Ma entriano nel vivo dell'argomento.

Come si è già visto, la possibilità di visualizzare una immagine in formato Koala (o altro), è legata unicamente ad una serie di trasferimenti di blocchi di memoria.

E, quando si tratta di spostare migliaia di byte, il linguaggio macchina la fa da padrone.

Per toccare con mano la differenza, si provi a digitare il listato 1.

Questo, presupponendo che esista un file Koala di nome "pic" sul dischetto in uso, ne effettua il caricamento (riga 40), quindi opera i trasferimenti necessari alla sua visualizzazione.

Il tutto strettamente in Basic... ed a vista.

Infatti, l'attivazione dell'hi-res (= le poke alle righe 50-70) è volutamente inserita all'inizio del programma, in modo da apprezzare esplicitamente la velocità con cui l'immagine va formandosi sullo schermo.

Dimostrazioni a parte, dedichiamoci ora ad una routine più "seria", scritta in Macro Assembler, che renda tutto più efficiente e, soprattutto, più veloce.

Si copi dunque il listato 2 e, non prima di averlo salvato, si impartisca Run.

Dopo una breve pausa, necessaria per allocare correttamente il codice LM memorizzato nelle linee data, un breve messaggio segnalerà che tutto è in ordine (salvo errori od omissioni), indicando le modalità di utilizzo della routine.

```
10 REM-------------------------------
20 REM          LISTATO  1
30 REM-------------------------------
40 IFA=0THENA=1:LOAD"PIC",8,1
50 POKE 53272,PEEK(53272)OR8
60 POKE53265,PEEK(53265)OR32
70 POKE53270,PEEK(53270)OR16
80 FORX=0TO7999
90 POKE8192+X,PEEK(24576+X)
100 NEXT:FORX=0TO999
110 POKE1024+X,PEEK(32576+X)
120 POKE55296+X,PEEK(33576+X)
130 NEXT:POKE53281,PEEK(34576)
140 END
```

In pratica, si inserisca nel drive un dischetto contenente una schermata Koala (rinominata), dopodichè si impartisca Sys 49152, 0.

Il programma chiederà il nome del file da caricare: lo si digiti, seguito da Return.

Il video scomparirà, il drive inizierà a ronzare e, giusto il tempo di ultimare il caricamento, la schermata verrà visualizzata sullo schermo.

Per tornare alle normali modalità grafiche, basterà premere un tasto qualsiasi.

Se in quest'ultima fase dovesse apparire un Syntax Error, non spaventatevi: la cosa è ininfluente e comunque, tra breve, si capirà come ovviare.

Un'altra possibilità della routine è quella di creare, sullo stesso dischetto in cui è presente il file Koala, tre nuovi file di nome "bitmap", "color2" e color1".

Sarà sufficiente, a tale scopo, attivare il programma LM con...

Sys 49152,1

...e, fornito il nome del file come per la modalità precedente, dopo il caricamento e la visualizzazione (tanto per rendere meno gravosa l'attesa) la routine provvederà alla memorizzazione su disco dei suddetti file.

Questi conterranno i già descritti elementi necessari al C64 per visualizzare una schermata, ma pronti per l'uso.

In parole povere, basterà caricarli in memoria con Load...,8,1 (ed attivare l'hires), e l'immagine sarà bella e servita.

Se quest'ultima procedura non dovesse risultare del tutto chiara, si faccia riferimento al listato 3 il quale, purchè siano già stati creati i tre file, provvede appunto ad utilizzarli in puro ambiente Basic, ma con una nuova velocità.

Si noti che, per attendere la pressione di un tasto, in riga 65 si fa ricorso al comando Wait, e non a Get.

Il motivo è presto detto: poichè non si sono modificati i puntatori del Basic per proteggere l'area da 8192 in poi, l'uso di variabili stringa (come, appunto, Get A$) potrebbe "sporcare" la pagina grafica.

A buon intenditor...

Prima di esaminare lo sviluppo Assembly della routine, è necessaria una precisazione.

La procedura è in grado di accorgersi di eventuali errori nell'accesso al disco, ma in maniera piuttosto grossolana.

Se, per esempio, non dovesse trovare il file da caricare, si limiterà a tornare in modo testo, mentre la

ANCHE CON IL C/128

La routine proposta in queste pagine può "girare" anche sul C/128, ma solo se attivato in modo 64.

Tuttavia i file che essa produce sono utilizzabili anche in modo 128, e con semplicità addirittura disarmante.

Infatti, quando con il C/128 si entra in hi-res, la matrice colore 1, ovvero quella video, viene trasferita dal sistema a partire da 7168, immediatamente prima della bitmap, che risiederà da 8192 in poi.

Il tutto, quindi, all'interno dell'area che, dopo un comando Graphic, viene riservata alla grafica.

Per adoperare le schermate Koala trattate dalla nostra routine, basterà quindi caricare il file "bitmap" alle sue già stabilite locazioni, quindi caricare da 7168 in poi la mappa colore 1, ed infine allocare anche il file "color2" allo stesso indirizzo che viene utilizzato con il C/64 (da 55296 in poi).

Unica particolarità (vedi numero scorso della rivista) da tenere presente, il fatto che il multicolor adopera l'area di memoria colore posta "sotto" quella standard, per cui occorre prima farla "emergere" manipolando la locazione 1.

Il colore di sfondo potrà essere prelevato dalla locazione 56296 (si sta parlando di banco 15, ovviamente).

In pratica, disponendo dei tre file generati dalla routine in modo 64, per visualizzare un'immagine Koala in modo 128 basterà utilizzare un listato come quello che segue:

```
10 graphic# 3, 1: bload "bitmap"
20 bload "color1", p 7168
30 poke 1, peek (1) and 254
40 bload "color2"
50 poke 1, peek (1) or 1
60 poke 53281, peek (56296)
```

Per non mostrare il formarsi della schermata, può essere inserito il comando "Fast" come prima istruzione, non dimenticando, ovviamente, di porre "Slow" dopo la linea 60.

spia del drive lampeggerà (più o meno furiosamente).

La sys di attivazione, però, può essere impartita dall'interno di un programma, per cui è più corretto usare qualche riga tipo...

```
10 open 15, 8, 15: sys 49152, 0
20 input#15, a, b$: print b$
```

...per essere sempre informati sugli incidenti di percorso.

Così facendo, tra l'altro, si eviterà l'eventuale (innocuo) Syntax Error di cui si è prima fatto cenno.

Il perchè, ce lo spiegherà il disassemblato MacroAssembler.

A TUTTO ASSEMBLY

Molto facile(!) e lineare risulta il listato assembly, che può essere ben esaminato grazie all'abbondanza di commenti posti a lato delle singole istruzioni.

La routine vera e propria inizia alla riga 40, essendo le precedenti dedicate alla solita inizializzazione delle varie label, nonchè alla definizione di due Macro (righe 20-38), che vedremo all'opera tra breve.

Le righe da 40 a 46 si occupano di prelevare il dato numerico dopo la sys, sfruttando l'ormai consueta routine di sistema $E200 (vedi numero precedente della rivista).

E' ovviamente necessario disporre di una schermata hi-res per attivare la rotine proposta

FIGURA 1

KOALA		
dec		hex
34576	col.fondo	$8710
34575	seconda matrice	$870F
33576	colore	$8328
33575	prima matrice	$8327
32576	colore	$7F40
32575		$7F3F
	bitmap	
24576		$6000

DOODLE		
dec		hex
32575		$7F3F
	bitmap	
24576		$6000
24551	mappa colore	$5FE7
23552		$5C00

PRINT SHOP		
dec		hex
24448		$5F80
	bitmap	
16384		$4000

Tutti gli appassionati posseggono i programmi grafici di cui si parla nell'articolo

Se il valore è diverso da zero o uno, si interrompe la procedura con un'uscita ad Illegal Error (46).

Le righe successive, da 48 a 69, ripropongono anch'esse qualcosa di già visto (n. 61): viene stampato sullo schermo il messaggio presente alle righe 157 e 158; in seguito viene abilitata la routine di Input del Basic per accettare da tastiera il nome del file Koala da caricare in memoria.

Viene poi stabilita la lunghezza del nome del file (57-60), in modo da adoperarla per Setname, una delle due routine del Kernal necessarie per effettuare un Load (se ne è parlato abbondantemente sempre sul numero 61).

Prima di richiamare anche la seconda (Setfile, righe 67-69), viene impostato a zero il registro Y, in modo che il Load venga poi effettuato all'indirizzo che più ci aggrada.

In particolare faremo (righe 76-77) iniziare il caricamento da 8192, ottenendo così una bitmap già correttamente allocata, operazione che ci risparmierà un ulteriore trasferimento.

Prima del Load vero e proprio, viene però disattivato lo schermo (71-73), più che altro per motivi estetici.

A questo punto, in memoria è presente tutto il necessario per visualizzare la schermata.

Non rimane dunque che trasferire le due mappe-colore che si trovano subito dopo la bitmap.

Lo scopo è raggiunto tramite la subroutine di riga 145 (anch'essa più che sfruttata nei precedenti appuntamenti con questa rubrica), che viene richiamata due volte (82-90 e 92-100) con i puntatori agli indirizzi di partenza ed arrivo settati di conseguenza.

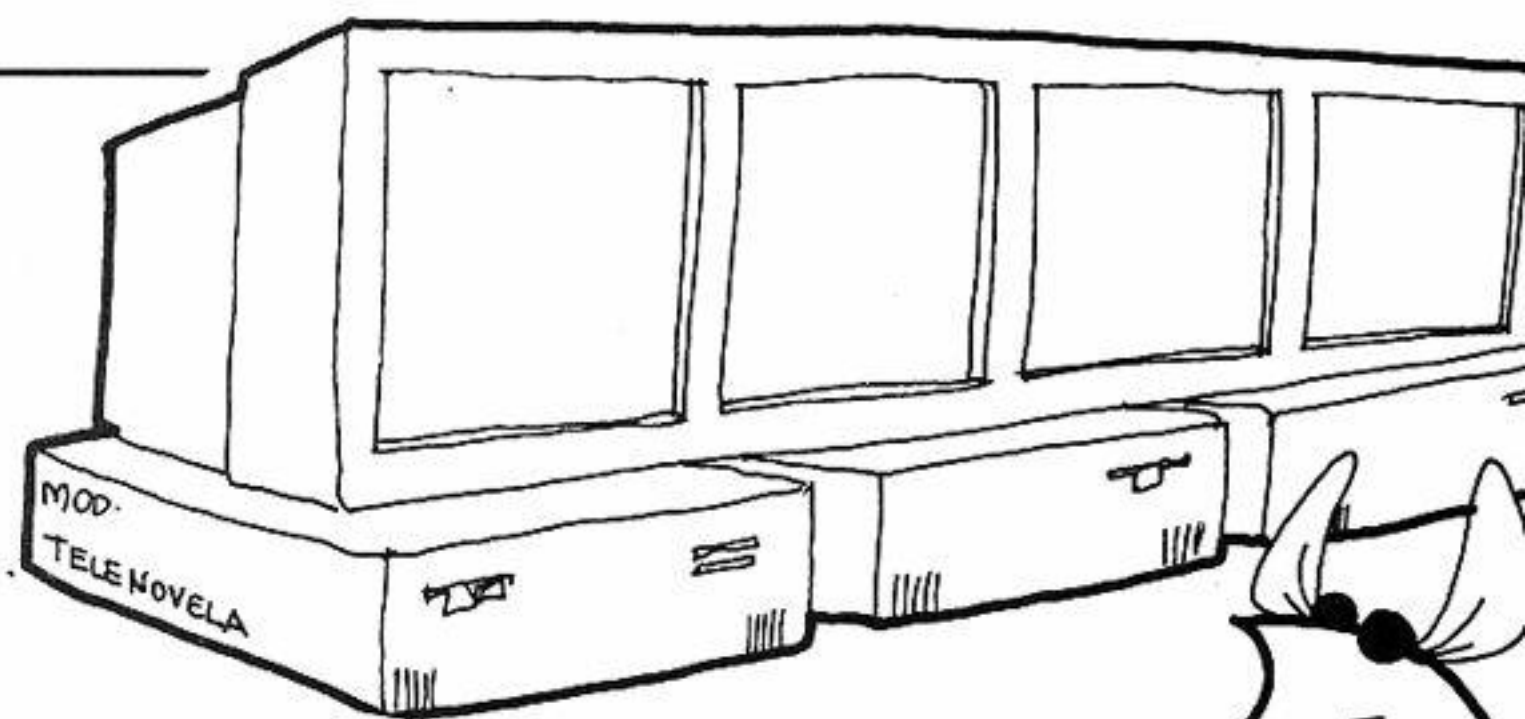
Infine, con le righe da 106 a 114, viene abilitata l'alta risoluzione multicolor ed eliminato il blank di schermo, provocando l'immediata comparsa, sul video, della schermata.

La routine controlla ora la locazione 2, dove era stato depositato il parametro della sys.

Se questo era zero, il programma entra in un loop che attende la pressione di un tasto (118-121); su-

La situazione si definisce (in HR): il comandante incarica Primo Giovedini di coordinare il lavoro dei tecnici che stanno cercando di levare la nave dalla scomoda situazione in cui versa.
Primo, aiutato dal buon Jack Assembler, inizia il lavoro...

Hei, Primo, non mi sembra che questi ragazzi si impegnino molto...!

Guarda questo qui: sta giocando a "SPACE INVADERS"!!!
ZIP! ZIP!
...scarsissimo!

```
100 REM ---------------------------------------------------------------
110 REM                     L I S T A T O   2
120 REM          TRATTAMENTO DELLE SCHERMATE KOALA(PAINTER)
130 REM ---------------------------------------------------------------
140 :
150 PRINTCHR$(147)"ATTENDI UN MOMENTO...":PRINT:PRINT
160 FORX=0TO332:READA:POKE49152+X,A:B=B+A:NEXT
170 IFB<>43999THENPRINT"ERRORE NELLE RIGHE DATA!":END
180 PRINT"SYS49152,0 - VISUALIZZA SCHERMATA KOALA"
190 PRINT:PRINT"SYS49152,1 - TRASFORMA SCHERMATA KOALA"
200 PRINTTAB(13)"IN FILES USABILI DA BASIC"
210 :
220 DATA 032,000,226,134,002,202,240,006,232,240,003,076,072,178
230 DATA 162,000,189,049,193,240,006,032,210,255,232,208,245,032
240 DATA 253,171,162,000,189,000,002,240,003,232,208,248,138,162
250 DATA 000,160,002,032,189,255,162,008,160,000,032,186,255,173
260 DATA 017,208,041,239,141,017,208,169,000,170,160,032,032,213
270 DATA 255,144,003,076,255,192,162,064,169,063,134,251,133,252
280 DATA 162,000,169,004,134,253,133,254,032,030,193,162,040,169
290 DATA 067,134,251,133,252,162,000,169,216,134,253,133,254,032
300 DATA 030,193,173,232,219,141,033,208,141,032,208,173,024,208
310 DATA 009,008,141,024,208,173,017,208,009,048,141,017,208,173
320 DATA 022,208,009,016,141,022,208,165,002,208,009,032,228,255
330 DATA 201,000,240,249,208,095,169,000,032,144,255,169,006,162
340 DATA 059,160,193,032,189,255,169,000,133,251,169,032,133,252
350 DATA 169,251,162,064,160,063,032,216,255,165,144,208,060,169
360 DATA 006,162,065,160,193,032,189,255,169,000,133,251,169,216
370 DATA 133,252,169,251,162,255,160,219,032,216,255,165,144,208
380 DATA 030,169,006,162,071,160,193,032,189,255,169,000,133,251
390 DATA 169,004,133,252,169,251,162,000,160,008,032,216,255,165
400 DATA 144,208,000,169,200,141,022,208,169,027,141,017,208,169
410 DATA 021,141,024,208,169,014,141,032,208,169,006,141,033,208
420 DATA 032,068,229,076,174,167,162,004,160,000,177,251,145,253
430 DATA 200,208,249,230,252,230,254,202,208,242,096,078,079,077
440 DATA 069,032,070,073,076,069,000,066,073,084,077,065,080,067
450 DATA 079,076,079,082,050,067,079,076,079,082,049
460 END
```

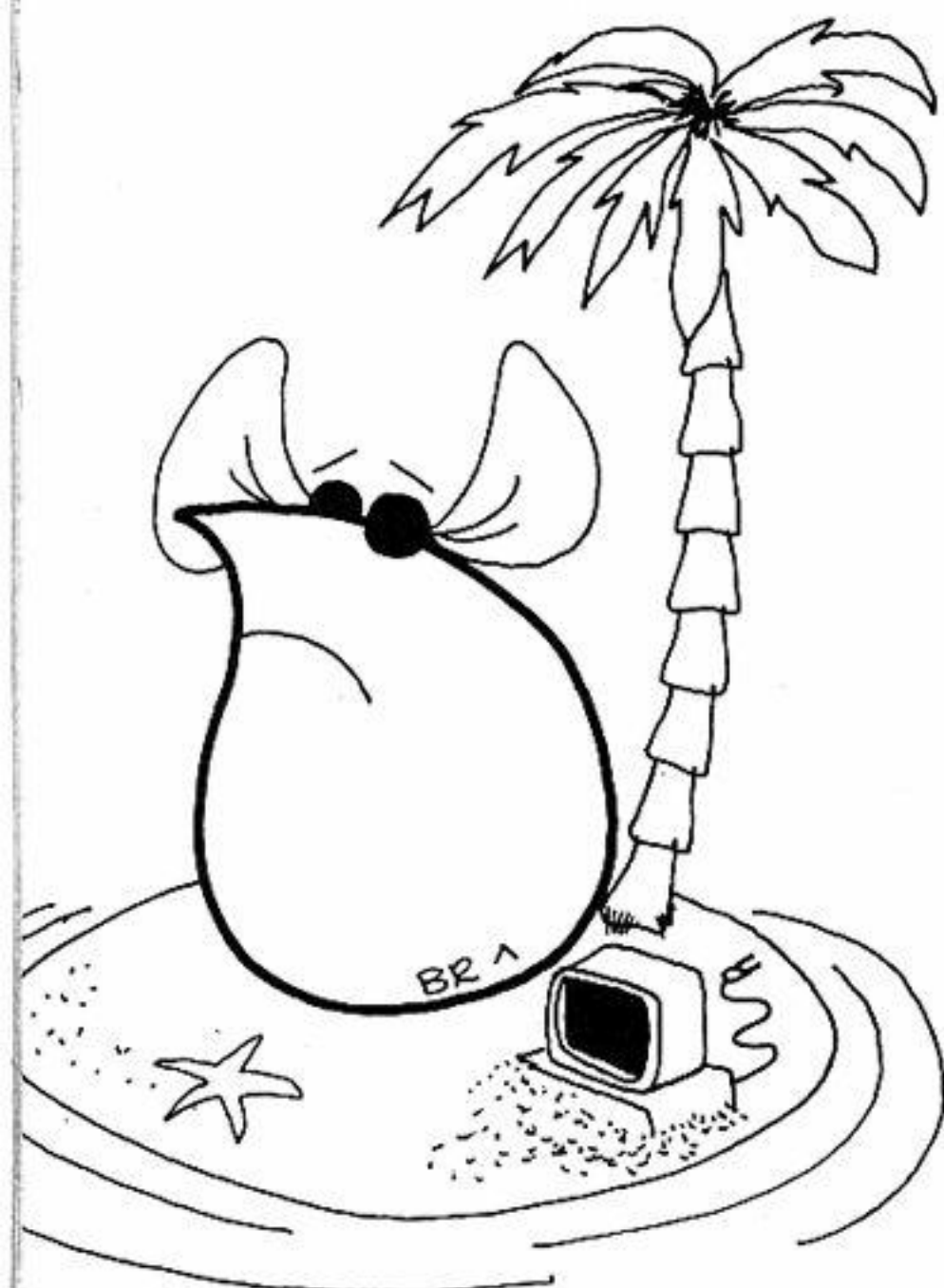



FIGURA 2

REGISTRI DEL VIC ADOPERATI NELLA ROUTINE

Registro	Bit	azzerato	settato
53265	bit 4	Blank ON	Blank OFF
	bit 5	Bitmap OFF	Bitmap ON

Registro	Bit	azzerato	settato
53270	bit 4	Multi color OFF	Multi color ON

Registro	Bit	Funzione
53272	bit 4-7	Indirizzo della matrice video. 0001 = default

Registro	Funzione
53280	Colore del bordo
53281	Colore del fondo

bito dopo salta alla sezione che si occupa di ripristinare i valori di default dello schermo e disattivare l'hi-res multicolor (132-143).

Il ritorno al Basic, come si può notare in riga 143, è effettuato tramite la routine di esecuzione del successivo comando (Execute Basic), in modo da sfruttare la Sys all'interno di un programma.

Se, però, la stessa viene data in modo diretto, ecco che la virgola ed il successivo parametro generano il Syntax Error di cui sopra.

Volendolo evitare (solo per il modo diretto!), è sufficiente modificare l'indirizzo di ritorno in $A474, la routine di Ready.

La cosa può essere fatta con il monitor del MacroAssembler, direttamente in fase di editing, e perfino da Basic con...

Poke 49436, 116: Poke 49437, 164

Tornando al disassemblato, le righe da 123 a 130, che entrano in ballo solo se il valore dopo Sys 49152 è uguale ad uno, provvedono a salvare su disco le tre aree di memoria che interessano l'alta risoluzione in multicolor.

In pratica vengono adoperate tre volte le Macro istruzioni (vedi n. 61 della rivista) di riga 20 e 27, che sfruttano le routine Setfile e Save del Kernal.

Il parametro posto dopo Filnam (125, 127 e 129) si sostituirà al punto interrogativo della relativa Macro, così come avverrà per i due specificati dopo Salv (126, 128 e 130), fornendo al Kernal i tre diversi punti di inizio e fine della memoria da salvare.

Si noti come l'uscita, in caso di errore del Dos, è realizzata differentemente per il Load ed il Save.

Nel primo caso (79-80), se dopo Load si è verificato un errore, il bit di riporto del registro di stato (Carry) risulterà settato, per cui, con BCC, viene verificata l'assenza (generica) di errori nell'accesso al disco.

Per il Save, invece, viene testata la locazione 144 (righe 36-37), corrispondente alla variabile ST del Basic: se è diversa da zero, vuol dire che qualcosa non è andato per il verso giusto.

Ed è tutto.

Non resta, dunque, che darsi da fare col joystick: la schermata, infatti, qualcuno deve pur farla!

```
10 REM------------------------------
15 REM          LISTATO  3
20 REM------------------------------
25 POKE53265,PEEK(53265)AND239
30 IFA=0THENA=1:LOAD"BITMAP",8,1
35 IFA=1THENA=2:LOAD"COLOR1",8,1
40 IFA=2THENA=3:LOAD"COLOR2",8,1
45 POKE53272,PEEK(53272)OR8
50 POKE53265,PEEK(53265)OR48
55 POKE53270,PEEK(53270)OR16
60 POKE53281,PEEK(56296)
65 POKE198,0:WAIT198,1
70 PRINTCHR$(147):POKE53281,6
75 PRINTCHR$(154)
80 POKE53272,PEEK(53272)AND247
85 POKE53265,PEEK(53265)AND223
90 POKE53270,PEEK(53270)AND239
95 END
```

DIO SALVI GLI SPRITE

Un'efficace routine l.m. permette di emulare, con il C/64, una potente istruzione disponibile sul C/128

di **E. Fichera Douglas, Paolo Monti**

Gestire stringhe risulta più semplice che gestire sprite

E' ormai arcinoto che il C/64, pur possedendo ottime potenzialità in merito alla grafica ed al sonoro, ha il suo tallone d'Achille proprio nel Basic V.2 che dovrebbe gestirli.

L'handicap è stato risolto con la realizzazione del Basic V.7, implementato però solo sul C/128; i 64-isti, infatti, invidiano ai possessori di questo computer non tanto la maggior quantità di memoria RAM a disposizione (sicuramente molto utile), quanto la facilità con cui è possibile creare effetti sonori e grafici grazie al gran numero di istruzioni ad essi dedicati; una di esse è certamente quella chiamata SPRSAV.

La funzione dell'istruzione SPRSAV (abbreviazione di SPRITESAVE) è quella di trasferire il "contenuto" di uno sprite in una variabile stringa (o viceversa) e/o da uno sprite ad un altro; il compito del programma in l.m. presentato in queste pagine è proprio quello di emulare l'istruzione SPRSAV del C/128.

ISTRUZIONI PER L'USO

Dopo aver attentamente digitato il caricatore Basic ed averlo salvato, si potrà impartire il RUN e, se non si saranno commessi errori, dopo qualche secondo la macchina risponderà con il consueto READY. A questo punto si dispone di un'istruzione Basic in più: vediamo come usarla.

Le sintassi possibili sono tre, tante quante le funzioni disponibili. La prima consente di memorizzare l'immagine di uno sprite (che si presuppone già disegnato in precednza) in una variabile stringa; il formato dell'istruzione è...

Sys 49152, Num sprite, Var$

...in cui "Num sprite" è, intuitivamente, il numero dello sprite (compreso tra 1 e 8 e non tra 0 e 7!) che contiene l'immagine da assegnare alla variabile VAR$ (di tipo, lo ricordiamo, rigorosamente STRINGA) che d'ora in poi conterrà i dati dello sprite.

La seconda funzione è inversa della prima dal momento che visualizza, tramite uno sprite, l'immagine memorizzata precedentemente in una variabile; la sintassi è...

Sys 49152, Var$, Num sprite

I parametri sono i medesimi dell'istruzione precedente; naturalmente non è necessario che la variabile Var$ sia quella (o una di quelle) generate precedentemente dall'istruzione inversa. Il computer la eseguirà fedelmente ma il risultato sarà probabilmente di scarso valore artistico.

La terza ed ultima funzione disponibile consente di "travasare" un'immagine da uno sprite ad un altro; la sintassi è...

Sys 49152, Num spr 1, Num spr 2

...in cui "Num spr 1" è il numero dello sprite di partenza e "Num spr 2" è il numero dello sprite che visualizzerà l'immagine-copia. L'immagine dello sprite di partenza non viene modificata, mentre quella dello sprite di arrivo viene irrimediabilmente distrutta a meno che non la si sia preventivamente salvata in un'altra variabile con l'istruzione:

Sys 49152, num spr 2, var$

Poichè il C/64 dispone di 8 sprites, il parametro NUM SPRITE può assumere, lo ripetiamo, valori compresi tra 1 e 8; valori esterni al range citato causano un "Illegal quantity error"; naturalmente il parametro suddetto può essere rappresentato anche da una variabile numerica e le variabili in cui memorizzare le immagini degli sprites devono essere esclusivamente di tipo stringa, anche a matrici unidimensionali o multidimensionali (è quindi possibile far visualizzare numerosi fotogrammi incrementando semplicemente l'indice di una variabile). Se si tenta di usare una variabile stringa in luogo di una numerica (e viceversa) il computer risponde con il classico "Type mismatch error".

PERICOLI

Apparentemente l'istruzione può essere usata senza alcun pericolo, ma non è così. Infatti il programma trasferisce i dati degli sprites da un punto all'altro della memoria utilizzando, come riferimento, i puntatori agli sprites allocati da 2040 ($07F8) a 2047 ($07FF). Prima di utilizzare l'istruzione SPRSAV è quindi necessario farli puntare a zone sicure della RAM, lontane da quella usata dal programma Basic contenente l'istruzione stessa o dalle prime quattro pagine di memoria che, come è noto, sono costituite da locazioni vitali per il funzionamento della macchina. Una prundente manipolazione del Top di memoria, inoltre, si rivela provvidenziale per la sopravvivenza della procedura.

COME FUNZIONA

Il disassemblato del programma è ampiamente commentato e coloro che smanettano da tempo con il linguaggio macchina non dovrebbero avere difficoltà a comprenderne il funzionamento.

I più esperti si accorgeranno che alcune parti del programma sono superflue se l'istruzione SPRITESAVE viene usata da sola, ma sul C/128 è possibile usarla anche in coppia con le istruzioni SSHAPE e GSHAPE che permettono di memorizzare, e manipolare, zone di schermo in alta risoluzione; le istruzioni ora superflue (esse si limitano ad inserire in testa alla variabile 2 bytes di link) permetteranno la piena compatibilità con eventuali programmi futuri (realizzati dai sottoscritti) per emulare anche le istruzioni suddette.

La routine proposta fa intenso uso di subroutines, utilizzate solitamente dall'interprete Basic, che la rendono molto compatta (e quindi veloce). Fra tutte, quella che costituisce il nucleo di SPRITESAVE è allocata a partire da $A3BF: il suo compito consiste nel trasferire blocchi di dati da una zona all'altra della memoria. Come forse è noto, essa deve avere, come input, l'indirizzo di partenza della zona da trasferire (nei bytes $5F / $60), l'indirizzo di fine in $5A / $5B e la FINE della zona di destinazione nei bytes $58 / $59.

Se si desidera "svuotare" uno sprite, NON è sufficiente impartire le istruzioni...

A$="": Sys 49152, a$, num sprite

...come sembrerebbe intuitivo; la macchina "azzererebbe", infatti, soltanto i puntatori della variabile e lascerebbe intatti i bytes successivi; è dunque necessario dichiarare una variabile contenente 65 zeri (2 link + 63 bytes dati) ad esempio con le istruzioni...

For a=0 to 64: a$=a$ + chr$(0): next

Digitando quindi...

Sys 49152, a$, num sprite

...si otterrà l'effetto desiderato; se, però, la procedura appare macchinosa, potrete spegnere lo sprite interessato tramite la locazione 53269 ($D015): infatti tra uno sprite spento e uno acceso (ma "vuoto") non sussiste alcuna differenza.

Concludiamo ricordando che ogni variabile generata da SPRITESAVE è lunga 65 bytes: attenzione a non rimanere a secco di RAM.

```
0 REM***********************************
1 REM**     SPRITESAVE PER C/64       **
2 REM**      COPYRIGHT 1989 BY:       **
3 REM**        PAOLO MONTI &          **
4 REM**     E. FICHERA DOUGLAS        **
5 REM***********************************
6 FOR A=49152 TO 49383: READ B
7 C=C+B: POKE A, B: NEXT
8 IFC<>31895THENPRINT"ERRORE NEI DATI"
9 :
10 DATA032,253,174,165,122,072,032,158
11 DATA173,165,013,240,003,076,164,192
12 DATA104,133,122,032,218,192,134,251
13 DATA032,253,174,165,122,072,032,158
14 DATA173,165,013,208,053,104,133,122
15 DATA032,218,192,134,252,166,251,189
16 DATA248,007,032,203,192,133,095,132
17 DATA091,132,096,024,105,063,133,090
18 DATA144,002,230,091,166,252,189,248
19 DATA007,032,203,192,132,089,024,105
20 DATA063,133,088,144,002,230,089,076
21 DATA191,163,104,133,122,032,139,176
22 DATA133,073,132,074,166,251,189,248
23 DATA007,032,203,192,133,253,201,002
24 DATA016,001,136,198,253,198,253,132
25 DATA254,160,000,177,253,072,138,145
26 DATA253,200,177,253,072,169,024,145
27 DATA253,169,065,133,252,134,101,169
28 DATA252,133,100,032,082,170,160,001
29 DATA104,145,253,136,104,145,253,169
30 DATA020,145,073,096,104,133,122,032
31 DATA139,176,133,253,132,254,032,253
32 DATA174,032,218,192,134,252,160,001
33 DATA177,253,200,170,177,253,168,232
34 DATA208,001,200,232,208,001,200,138
35 DATA076,053,192,162,000,134,255,024
36 DATA106,102,255,106,102,255,168,165
37 DATA255,096,032,158,183,202,048,004
38 DATA224,008,144,003,076,072,178,096
39 END
```

FULL DI "ROBCOM"!

SCALA DI "NIKI"!

A! Vedo che state giocando a POKER, ragazzi...

Si, Primo, per l'esattezza...

...giochiamo a POKER 53265,0!

```
.;************************************
.;**     DISASSEMBLATO SPRITESAVE    **
.;**              di                 **
.;** E. Fichera Douglas & Paolo Monti **
.;** Copyright 1989 - fo Simy & Sabry **
.;************************************

,c000 jsr $aefd       ;prende una virgola     .
,     lda $7a         ;salva posizione        .
,     pha             ,carattere corrente     .
,     jsr $ad9e       ;valuta l'espressione.
,     lda $0d         ,e' una stringa ?       .
,     beq $c010       ;no: continua           .
,     jmp $c0a4       ;si':salta              .
,c010 pla             ;si posiziona per       .
,     sta $7a         ;leggere il parametro.
,     jsr $c0da       ;lo legge               .
,     stx $fb         ;e lo memorizza         .
,     jsr $aefd       ;prende una virgola     .
,     lda $7a         ;salva posizione        .
,     pha             ;carattere corrente     .
,     jsr $ad9e       ;valuta l'espressione.
,     lda $0d         ;e' una stringa ?       .
,     bne $c05a       ;si':salta              .
,     pla             ;no:si posiziona per .
,     sta $7a         ;leggere il parametro.
,     jsr $c0da       ;lo legge               .
,     stx $fc         ;e lo salva             .
,     ldx $fb         ;prende 1' parametro .
,     lda $07f8,x     ;legge puntat. sprite.
,     jsr $c0cb       ;calcola zona dati      .
                      ;sprite                 .
,c035 sta $5f         ;e ne memorizza         .
,     sty $5b         ;i puntatori            .
,     sty $60         ;                       .
,     clc             ;calcola e              .
,     adc #$3f        ;fissa                  .
,     sta $5a         ;fine zona              .
,     bcc $c044       ;dati sprite            .
,     inc $5b         ;                       .
,c044 ldx $fc         ;prende 2' parametro .
,     lda $07f8,x     ;legge puntat. sprite.
,     jsr $c0cb       ;calcola e memorizza .
,     sty $59         ;inizio dati sprite     .
,     clc             ;calcola e memorizza .
,     adc #$3f        ;fine                   .
,     sta $58         ;zona                   .
,     bcc $c057       ;dati sprite            .
,     inc $59         ;                       .
,c057 jmp $a3bf       ;trasferisce i dati     .
                      ;e ritorna al Basic     .
,co5a pla             ;si posiziona per       .
,     sta $7a         ;leggere la variabile.
,     jsr $b08b       ;la legge               .
,     sta $49         ;e ne salva             .
,     sty $4a         ;i puntatori            .
,     ldx $fb         ;legge il parametro     .
,     lda $07f8,x     ;legge puntat. sprite.
,     jsr $c0cb       ;calcola                .
,     sta $fd         ;e fissa                .
,     cmp #$02        ;inizio zona            .
,     bpl $c073       ;dati sprite            .
,     dey             ;                       .
,co73 dec $fd         ;                       .
,     dec $fd         ;                       .
,     sty $fe         ;                       .
,     ldy #$00        ;inserisce              .
,     lda ($fd),y     ;due bytes              .
,     pha             ;nella variabile        .
,     txa             ;per                    .
,     sta ($fd),y     ;'Future expansion'     .
,     iny             ;(vedi testo)           .
,     lda ($fd),y     ;                       .
,     pha             ;                       .
,     lda #$18        ;                       .
,     sta ($fd),y     ;                       .
,     lda #$41        ;fissa lunghezza        .
,     sta $fc         ;della variabile        .
,     stx $65         ;predispone             .
,     lda #$fc        ;i puntatori            .
,     sta $64         ;                       .
,     jsr $aa52       ;assegna la variabile.
,     ldy #$01        ;rimette a posto        .
,     pla             ;i due bytes            .
,     sta ($fd),y     ;originari              .
,     dey             ;                       .
,     pla             ;                       .
,     sta ($fd),y     ;                       .
,     lda #$14        ;                       .
,     sta ($49),y     ;                       .
,     rts             ;ritorna al Basic       .
,c0a4 pla             ;si posiziona per       .
,     sta $7a         ;leggere la variabile.
,     jsr $b08b       ;la legge               .
,     sta $fd         ;e ne salva             .
,     sty $fe         ;i puntatori            .
,     jsr $aefd       ;prende una virgola     .
,     jsr $c0da       ,legge il parametro     .
,     stx $fc         ;e lo salva             .
,     ldy #$01        ;non prende             .
,     lda ($fd),y     ;in considerazione      .
,     iny             ;i primi due bytes      .
,     tax             ;della variabile        .
,     lda ($fd),y     ;                       .
,     tay             ;                       .
,     inx             ;                       .
,     bne $c0c3       ;                       .
,     iny             ;                       .
,c0c3 inx             ;                       .
,     bne $c0c7       ;                       .
,     iny             ;                       .
,c0c7 txa             ;                       .
,     jmp $c035       ;trasferisce i dati     .
,c0cb ldx #$00        ;moltiplica             .
,     stx $ff         ;l'accumulatore X64     .
,     clc             ;                       .
,     ror             ;                       .
,     ror $ff         ;                       .
,     ror             ;                       .
,     ror $ff         ;                       .
,     tay             ;                       .
,     lda $ff         ;                       .
,     rts             ;e ritorna              .
,c0da jsr $b79e       ;legge un parametro     .
,     dex             ;e controlla che        .
,     bmi $c0e4       ;sia accettabile        .
,     cpx #$08        ;                       .
,     bcc $c0e7       ;                       .
,c0e4 jmp $b248       ;                       .
,c0e7 rts             ;ritorna
```

Il risultato è immediato...

BLANK!

Beh? Cosa succede?

Chi ha spento la luce?

Eravamo in alta risoluzione e ora non si vede più niente!

Primo si irrita...

Ecco, bravo! Dovevi proprio dare quella POKE?

Ora lo schermo è BLANKATO!

K-SEKA ASSEMBLER, PRIMO INCONTRO

Non si può sacrificare un computer come Amiga nei ristretti ambiti del Basic; cerchiamo di usare linguaggi più adeguati

di **Gli Amici di Amiga**

Presentiamo l'ottimo assemblatore della Andelos Systems distribuito dalla KUMA Computers, importante strumento per chiunque desideri controllare direttamente l'hardware del proprio Amiga.

Il Seka è un programma di appena 24K eseguibile direttamente da CLI; si crea una nuova finestra ed è pronto per essere utilizzato anche insieme ad altri task (mentre sto scrivendo questo articolo nella finestra del vecchio Notepad, ho sott'occhio il Seka che sta girando nella sua comoda finestra ridimensionabile).

Incorpora un text editor, un assembler, un disassembler, un debugger e un linker trasferiti in memoria permanentemente, così che non ci siano più accessi al disco (cosa che con altri assemblatori provocava stressanti pause), con conseguente risparmio di tempo e di energia cerebrale.

Una volta lanciato, compare il messaggio...

"Andelos 68000 Assembler SEKA v1.5 Copyright Andelos Systems Amiga Version"

...e l'assemblatore chiede la quantità di memoria (misurata in Kilobyte) con la quale si desidera lavorare; tenete conto che in essa dovranno essere contenuti sia il codice sorgente che il codice e il buffer per il linker; inoltre, se avete l'espansione RAM, il Seka si installerà in quest'ultima. Dunque cercate di non essere avari e, se avete bisogno di lavorare solo con la CHIP-RAM, usate il NoFastMem prima di caricarlo.

Impostata la quantità di memoria con cui lavorare (provate inizialmente con 30 K) apparirà il prompt "SEKA)33 e l'assembler è pronto a ricevere comandi.

IL TEXT-EDITOR

Con la semplice pressione del tasto ESC si entra (e si esce) nel Text-Editor grazie al quale si inserirà il listato sorgente.

Ogni linea può esser composta da:

- Numero di linea
- Label seguita dal carattere di doppio punto (quando serve!)
- Istruzione
- Operando
- Commento, preceduto da un punto e virgola.

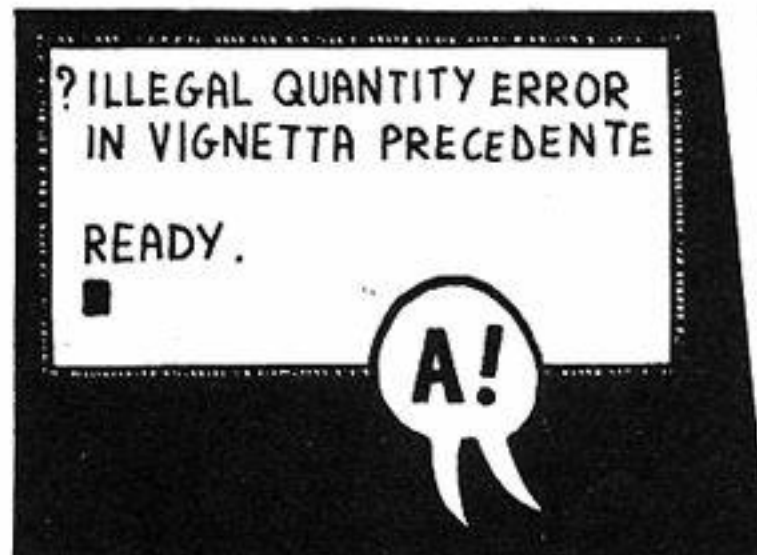
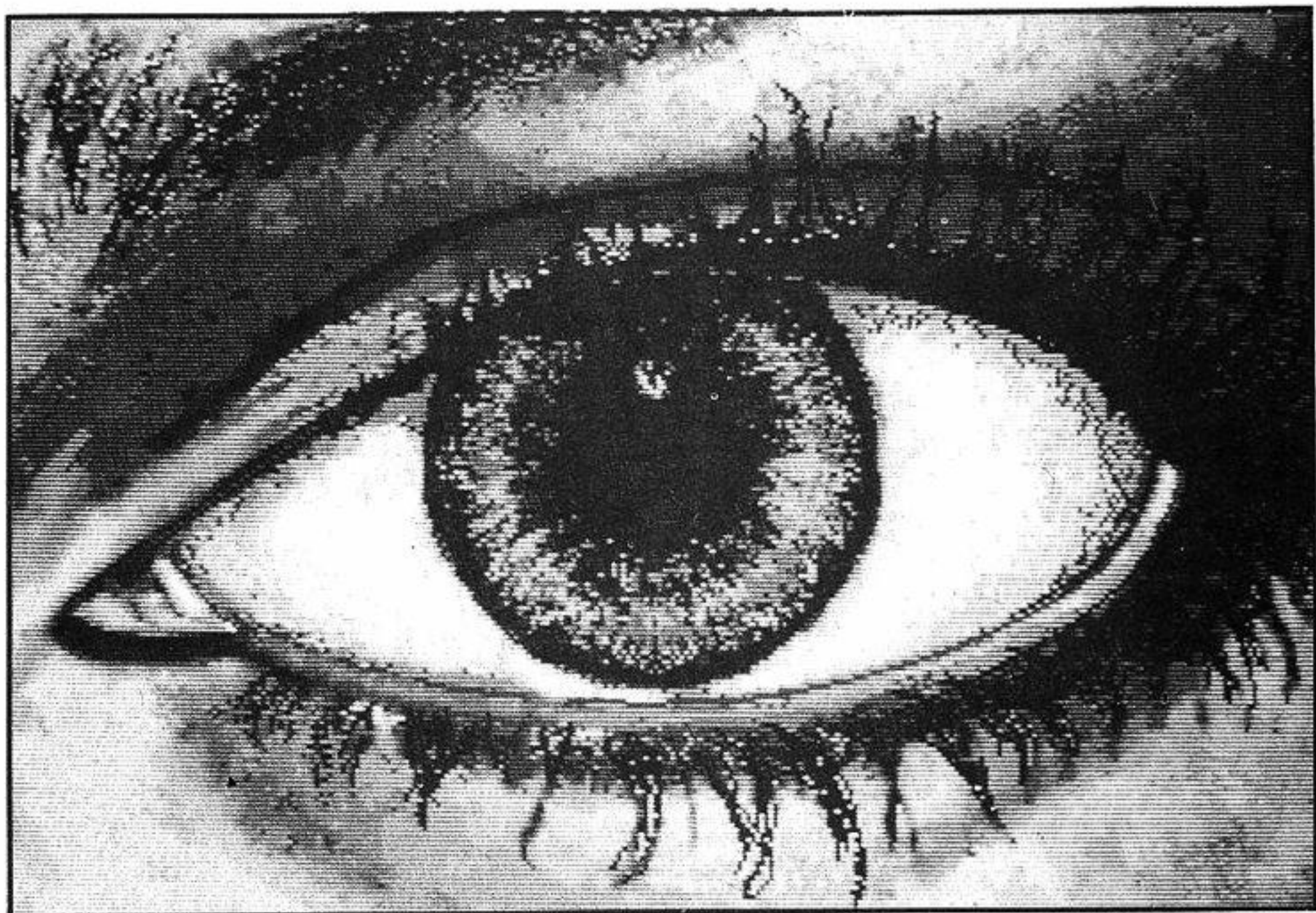
Se nella linea si trova solo un commento, questo deve essere preceduto da almeno un asterisco, oppure dal solito carattere di punto e virgola.

L'assemblatore è molto flessibile e non richiede particolari separatori tra i vari campi. Tuttavia, per chiarezza, consigliamo di utilizzare la tabulazione come in tutti i listati in assembly che saranno pubblicati in seguito.

E' possibile assegnare valori ai simboli (costanti) sia tramite la pseudo-istruzione EQU (nel qual caso il nome del simbolo deve essere seguito dai due punti), sia con il carattere eguale (=). I valori numerici esadecimali devono essere preceduti dal simbolo del dollaro ($), quelli ottali dal simbolo della chiocciola, mentre quelli binari dal simbolo di percentuale (%).

Per inserire in memoria un certo numero di valori numerici viene usata la pseudo-istruzione...

DC.x Valore1, Valore2,..., ValoreN

Procuratevi il semplice assemblatore SEKA

L'Assembly del 68000 è più semplice di quello del 6510

...in cui il suffisso ".x" specifica il tipo di dato e può essere uno dei seguenti:

.B bytes (8 bit)
.W words (16 bit)
.L long words (32 bit)

Se invece c'è bisogno di inserire in memoria una stringa di caratteri, ci serviremo della pseudo-istruzione:

DC.B "Questa è una stringa esempio"

Analogamente, per inserire in memoria un medesimo valore N volte, useremo la pseudo-istruzione:

BLK.x N, Valore

Se una parte del codice deve essere assemblata a partire da una determinata locazione di memoria, basterà farla precedere dalle seguenti due istruzioni:

ORG indirizzo
LOAD indirizzo

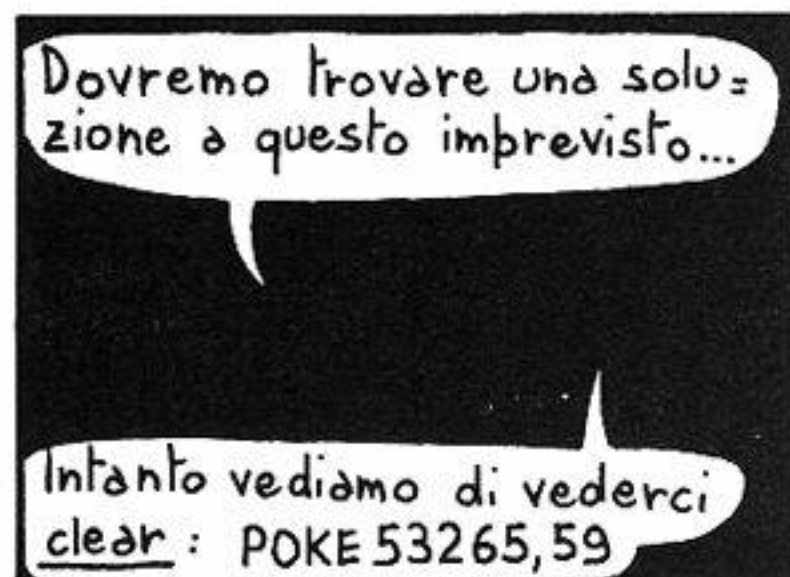
in cui "indirizzo" è un numero pari.

Il Text-Editor è dotato della funzione di Cut & Copy: dopo aver portato il cursore sul carattere di inizio del blocco si preme CTRL-B, lo si posiziona sul carattere di fine del blocco e si preme CTRL-C.

Il blocco verrà cancellato per essere poi inserito a partire dalla locazione corrente del cursore con la pressione di CTRL-P.

Per uscire dall'Editor si usa nuovamente il tasto ESC.

ASSEMBLAGGIO E ESECUZIONE

Siamo ora pronti a digitare il piccolo programma di prova di figura 1 che individua il valore maggiore

AMIGASSEMBLY

Da tempo era stata presa la decisione di ripetere, con Amiga, ciò che un anno fa iniziavamo a sviluppare per il C/64.

Parlare di Assembly, tuttavia, è un'impresa facile e difficile nello stesso tempo. Da una parte il 68000 consente un'elasticità decisamente superiore al 6510; dall'altra c'è da affrontare il problema della comprensione dell'architettura propria di Amiga.

Non ultima è da sottolineare la difficoltà di parlare di programmi assemblatori facilmente rintracciabili dal lettore "medio".

L'anno scorso la nostra scelta era caduta su "Macroassembler", di casa Commodore, sicuri di una sua facile reperibilità. Le osservazioni di alcuni lettori (che lamentavano la carenza dell'ottimo dischetto originale perfino presso i Commodore Point) ci ha fatto riflettere sulla reale opportunità di favorire un nome piuttosto che un altro.

La nostra scelta è quindi caduta sul popolare e diffusissimo SEKA, certamente a disposizione di tutti coloro che (come l'autore di questo primo articolo, **Daniele Paccalone**) smanettano con Amiga.

Sappiamo benissimo che esistono molti altri (e più potenti) assemblatori, regolarmente importati perfino in Italia. Ma dopo i primi contatti con alcuni imporatori ci siamo dovuti arrendere di fronte alla indifferenza degli stessi a sostenere un segmento di mercato (quello degli smanettoni, appunto) certamente meno redditizio di quello, ben più vasto, dei giocatori ammazzamarzianetti.

E allora, viva i pirati, grazie ai quali vengon fatti circolare programmi (e notizie sul loro uso) che non vedrebbero la luce se affidati eslusivamente ai mercanti di videogames.

A.d.S.

tra i dieci della lista posta in fondo al programma e lo trascrive nel registro D0.

Dopo averlo digitato, usciti dall'Editor, possiamo assemblarlo con il comando "A". Al prompt "OPTIONS)33 rispondete con un semplice RETURN perchè, almeno per il momento, non avete bisogno di particolari opzioni (che vedremo in prossimi articoli).

Se vengono segnalati errori ricontrollate il listato sorgente, altrimenti eseguitelo (dopo averlo salvato!) tramite il comando "J indirizzo", dove indirizzo è una espressione indicante l'indirizzo iniziale del codice.

Nel nostro caso avendo introdotto all'inizio del programma la label "s:", (e per semplificare le cose) basterà digitare "J s".

Terminata l'esecuzione, il Seka visualizza tutti i registri, tra i quali è anche il nostro D0 che conterrà il valore cercato in rappresentazione esadecimale.

Per visualizzarlo in forma decimale si usa il comando "? espressione", dove espressione può essere anche un registro.

Cioè:

```
? $0000E290
```

oppure

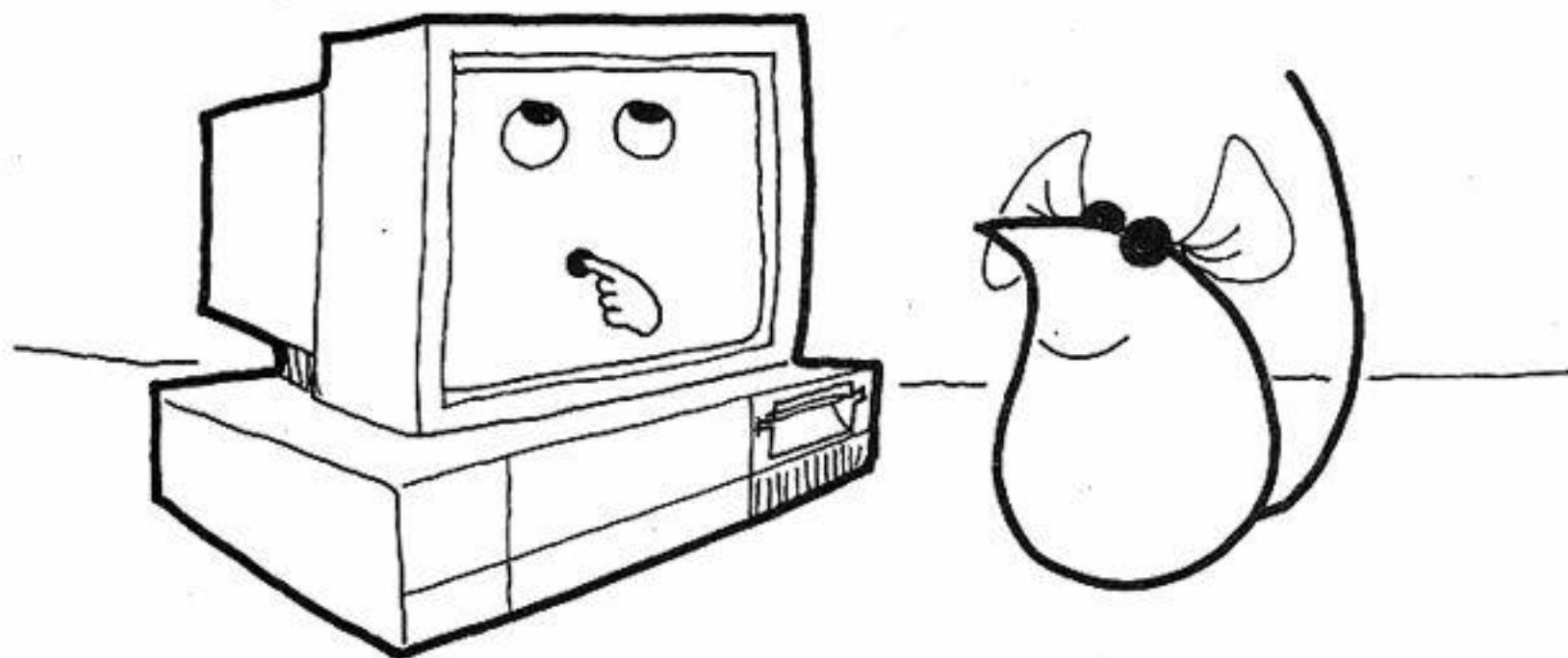
```
? D0
```

Se volete eseguire il programma passo-passo per analizzare i registri dopo ogni istruzione, usate il comando "S indirizzo".

I due semplicissimi listati vi aiuteranno a compiere i primi passi

GESTIONE DEL DRIVE

I principali comandi per la gestione del drive sono:

```
*** MAGGIORE (Amiga + Seka) ***

NumValori = 10

s:      move.b #NumValori,d2      ;d2 fa da contatore.
        move.l #lista,a0          ;a0 punta al primo elemento della
                                  ; lista.
        clr.l  d0                 ;Azzera il registro d0.

test   :cmp.w  (a0),d0            ;Compara l'elemento puntato da a0 con
                                  ; d0.
        bhi.s   NonMag            ;Se non e' maggiore salta la prossima
                                  ; istruzione...
Mag    :move.w (a0),d0            ;Metti in d0 il nuovo valore, essendo
                                  ; maggiore del precedente.
NonMag:subq.b  #1,d2              ;Decrementa il contatore.
        beq.s   Termin            ;Se era l'ultimo valore esci...
        addq.l  #2,a0             ;Fai puntare ad a0 il prossimo valore
                                  ; della lista.
        bra.s   test              ;Ritorna al test...

Termin: rts                       ;Esci.

* Questa e' la lista dei 10 valori (occupanti ognuno una word).
* I numeri devono essere >0 e < 65535

lista:
dc.w    345, 233, 5678, 3444, 2, 9555, 2000, 58000, 10000, 8034
```

```
**** LED & FILTER ON/OFF (Amiga + Seka) ****
s:
  bchg   #1,$bfe001      ;Inverte lo stato del bit 1.
  rts                    ;Esce.
```

V
Visualizza il contenuto della directory e opera l'equivalente del comando CLI "CD directory". Per tornare alla directory "parent" basta ripetere esattamente lo stesso comando.

W
Salva il sorgente su disco chiedendone il nome e aggiungendo il suffisso ".s".

KS
Cancella il programma in memoria

R
Carica il sorgente da disco chiedendone il nome e aggiungendo il suffisso ".s" Ricordate, prima di caricare il nuovo programma, di cancellare quello vecchio con il comando "KS" altrimenti quest'ultimo verrà aggiunto in coda.

WO
Salva il programma oggetto su disco chiedendone il nome. Il programma potrà così essere eseguito da CLI.

WI
Salva su disco una zona di memoria.

RI
Carica da disco in memoria (p.e. un'immagine).

KF
Cancella un file.

DEBUG

Ecco la lista dei principali comandi di Debug:

X
Esamina registri.

G address

Esegue il programma.

J address
Salta alla subroutine.

Q address
Esamina memoria.

N address
Disassembla.

M address
Esamina e permette di modificare locazioni contigue.

S address
Esegue il programma passo-passo.

F
Fill di una zona di memoria.

C
Copy di una zona di memoria.

? espressione
Mostra (in esadecimale o decimale) il valore dell'espressione.

UN UTILE PROGRAMMA

Il secondo programma ha dimensioni alquanto ridotte; è infatti composto da due sole istruzioni, compresa la RTS per terminarne l'esecuzione.

Il sistema audio di AMIGA è costituito da quattro convertitori D/A, che trasformano i dati della forma d'onda, conservati in memoria come bytes, in segnali analogici da mandare ad un impianto di amplificazione.

Il segnale d'uscita viene però tagliato da un filtro che, a partire da 4 KHz, elimina gradualmente tutte le frequenze superiori.

La procedura, automatica, serve per eliminare le pseudo-frequenze introdotte a causa delle limitazioni dei convertitori, al di sopra di 13 KHz.

Un filtro che taglia le frequenze a partire da 4 KHz rappresenta una notevole limitazione se si pensa che, se i suoni sono stati campionati correttamente, le pseudo-frequenze sono ridotte a livelli minimi.

Nei più recenti modelli di AMIGA (2000B e la maggior parte dei 500) è possibile disattivare il filtro per ottenere l'intera gamma di frequenze fino alla massima umanamente udibile (20KHz).

Il bit di controllo del filtro è lo stesso che controlla il LED "power": se la luce è accesa sarà attivo il filtro, e viceversa. Esso è il bit 1 della locazione $BFE001.

Su AMIGA 2000 (e su alcuni modelli A-500) il LED non si spegne ma emette una luce di minore intensità.

Che cosa fa il nostro programma?

Cambia semplicemente lo stato del bit per mezzo dell'istruzione BCHG (Bit-CHanGe). Facendo girare una seconda volta lo stesso programma si otterrà la condizione di partenza (si pensi all'interruttore del monitor o del televisore).

Ora vedremo come trasformare il comando SEKA in un comando CLI.

IN MODO CLI

Dopo averlo assemblato, salvatelo con il comando WO precedentemente descritto: alla richiesta del nome del file digitate "LedFilt" o un qualsiasi altro nome di vostro gradimento.

Se tutto è andato per il verso giusto troverete sul disco il nuovo comando eseguibile direttamente da CLI.

Volendo lo si può ricopiare su qualsiasi disco e magari inserirlo nella Startup-Sequence in modo da avere il filtro subito disattivato, cosa molto utile nella maggior parte dei programmi musicali.

Come già detto, per riattivare il filtro (e di conseguenza il LED), basterà far girare nuovamente il programma.

Allo scopo di verificare l'effettivo funzionamento del programma, agite come segue:

• Attivate (da workbench), il comando "Say" presente nel cassetto "Utilities" e fate pronunciare più volte una frase qualsiasi, sufficientemente lunga.

• Aprite la finestra CLI (oppure Shell, se avete il W.b 1.3).

• Lanciate il programma "ledfilt" di cui abbiamo già parlato, e verificate che la luce del LED cambi la sua intensità.

• Cliccate sulla finestra di "Say" (Input Window) e fate ripetere la stessa frase di prima: dovreste notare un'accentuata differenza.

• Eseguite più volte la procedura appena descritta per meglio apprezzare la differenza di pronuncia.

Inutile dire che anche da Basic, facendo eseguire comandi "sonori", sarà possibile accorgersi della presenza, o meno, del filtro di frequenza.

Vuoi pubblicare un annuncio gratuito?

Commodore Computer Club ti offre la possibilità di pubblicare GRATUITAMENTE il tuo annuncio riguardo scambio o vendita di software, vendita o acquisto di apparecchi usati, ricerca di amici per fondare un club e così via. Invia l'annuncio in busta chiusa, affrancata secondo le norme vigenti, indirizzando a:

Commodore Computer Club - Viale Famagosta, 75 - 20142 Milano

Cerco stampante grafica MPS 803, funzionante, a prezzo modesto, possibilmente in zona Milano. Andreoni Damiano, Via C. Colombo n. 20 - 20083 Gaggiano (Mi). Telefonare o scrivere: 02/9085800

Scambio programmi e manuali per Amiga, o vendo a L. 3.000 dischetto compreso (manuali L. 200 a fotocopia). Cerco manuale di "Provideo Plus". Qualche Amigo mi può aiutare? Mauro Bricca, Via Monade 38 - 18013 Diano Marina (Im). Tel. 0183/400814 oppure 495491

Vendo C64 + Drive 1541 + registratore C2N + 2 Joystick + 2 Paddle + Speeddos Plus + adattatore telem. 6499 + varie cassette + istruzioni manuali, libri e riviste a L. 900.000. Maurizio Piraccini, Viale Dei Pini n. 8 - Cesena. Tel 0547/26958

Vendo Commodore 128 + manuali + Disk Drive 1571 + registratore e circa 500 programmi a lire 850.000. Cuorgnè- Torino. Tel. 0124/667093

Causa passaggio ad Amiga vendo PR4 su disco per CBM 64 (oltre 700 titoli tra Games & Utility) a L. 5.000 il disco (2 facciate) vendo inoltre enciclopedia basic "Input" in 6 volumi per imparare a programmare su CBM-Spectrum e altri mini a L. 80.000. Tel. 041/4966723

Cerco programmi su disco d'astronomia, per C64 (anche pirati). Per saperne di più telefonare allo 0594/84053 o scrivere a: Pellegrini Michele, Via Giovan Battista Vico n. 12 - Forte Dei Marmi.

Attenzione!!! Geos64? Therès no problem! A sole L. 15.000 + supporto potrete avere il nuovo Geosy1.3. Non siete soddisfatti? E allora ecco Blazing Paddle! Print Shop! Koala! Desktop publishing! Copiatori (ben 2 dischi) a prezzisuper. Telefona allo 0422/60847 o scrivi a: Schievano Mauro, Via Nervesa 7C - 31100 Treviso.

Vendo giochi su cassetta X CBM64 (190 giochi su 22 cassette) + registratore nuovo di 1 mese, a L. 150.000 (cassette d'edicola e non). Claudio Portelli, P.za Bugiardini n. 2 - 63012 Cupra Marittima (Ascoli P.). Tel. 0735/777756

Per passaggio ad Amiga, vendo C64 nuovo modello + registratore orig. + Joystick: in omaggio alcuni giochi. Imballi originali, pochi mesi di vita. Roberto: tel. 6252175 (ore pasti - solo Roma).

Vendo adattatore telematico Mod. 6499 per C64 nuovo mai usato a L. 150.000. Per informazioni telefonare a: Fasanella Michele, Via G. Pascoli 8 - 20040 Agliate Br. (Mi). Tel. 0362/990299

Vendo C64 + Drive 1541 + registratore + 400 giochi selezionati + prg. fantastici con relative guide, tutto in ottime condizioni L. 600.000. Davide: telefonare allo 0542/672768

Cerco/compro floppy disk vuoti da 5'25", preferibilmente 2 Sided, 2D per un prezzo non superiore a L. 900 l'uno. Toma Pasquale, Via G. Palmieri - Zona industriale Tuglie (Le).

Cerchiamo soci per il nuovo Club C64. Tutti gli iscritti riceveranno 4 giochi omaggio e ogni mese una rivista con le migliori novità!!! Pierangelo Galizia, Via Appia 3 - 85050 Baragiano Sc. (Pz). Tel. 0971/993258

Vendo C128 + Drive 1571 compatibile + penna ottica corredata di software + 1 Joystick ed inoltre regalo 80 dischi di giochi e utility con relativo raccoglitore. L. 650.000. Giuliano: tel. 091/588179

Amiga Club Napoli: acquisto, vendita, cambio software. Vasta disponibilità manuali. Informazioni: Massimiliano Afiero, C.so Italia Isolato 11 - 80021 Afragola (Na). Tel. 081/8697713

Vendo/scambio software per Amiga e C64 (posseggo i migliori programmi). Sorda Pasqualino, Via Napoli n. 25 - 82020 Fragneto Monforte (Benevento). Vi aspettotel. 0824/98636

Attenti!! Agli appassionati di musica e grafica Amiga. Vendo a prezzi bassissimi i migliori programmi. Posseggo Digi Paint, Delux Paint II, Fantavision, The Butcher, Aegis Sonix etc. Scrivere a Marco Giubelli, Via Alla Cartiera 14 - Serravalle Sesia (Vc).

Il Computer Club Vercelli esiste. Hai un computer (Ms-dos, Amiga, C64, Spectrum od altro) e cerchi amici con cui scambiare idee, opinioni, programmi? Vieni a trovarci sabato pomeriggio in Piazza C. Battisti n. 7 (Ex-Enal). Per informazioni telefonare in ore serali allo 0161/63074.

Cerco istruzioni dettagliate per giochi "Moon Ground" e "Simulatori di Volo 4" (logica 2000). Anche fotocopie. Contattarmi per prezzo: Luca Cassioli, Via Palombarese Km. 23, 500 - 00010 S.A. Romano (Rm). Tel. 0774/303010

Cerco software per C64. Tipo Spacearrier, Double Dragon, ecc. Scrivere a Marco, Via Montoggio 20 - 00168 Roma. (Sono disposto anche a scambiare programmi).

Alt! offro "tutti" i migliori copiatori e prg. Totocalcio-Lotto per C64/Amiga/IBM. Max serietà!!! Ascione Maurizio, Via Panoramica 10 - 80056 Ercolano (Na). Tel. 081/7392240 (Ore 16.00).

A.A.A. cerc. amici per fondare un Club. Solo possessori C64-128 e abitanti a Taranto e provincia. Pierluigi Natale, Via Tiziano 12 - 74027 S. Giorgio J. (Ta). Tel. 099/696809 (Ore 19.30).

A.A.A.A. Vendo programmi vari per qualsiasi comp. Spec. matematica, rubriche personalizzate. Prezzi modici. Gatto D'Arrigo Santi. Tel. 0982/612003

Il nostro Club C128 che è appena nato cerchiamo dei soci, e vendiamo delle cassette originali degli Atari 64k e qualche cassetta del C128, se volete associarvi al nostro club o comprare tel. tel al seguente numero: 0491/714423

Scambio software per Amiga 500 preferibilmente zona di Roma. Inviare la propria lista. Scrivere o telefonare dopo le ore 21.00. Riccardo Losciale, Largo Ugo Bartolomei, 5 - 00136 Roma. Tel. 06/3588397

Cerco la "Niki Cartridge" prezzo favorevole tra L. 40.000-45.000 scrivere a Maurizio Fischetti, Via Nazionale n. 11 - 82030 Dugenta (Bn).

Cerco programmi "copiatori" a nastro per C64 e di contabilità e di ingegneria civile sempre per 64. Peppe Sutera, Via Dante Alighieri 42 - 92013 Menfi (Ag). Tel. 0925/74659

Causa passaggio Sys. Sup. svendo A 500 (512 K + F.D. interno da 31/2") + A501 (espansione 512 K) + monitor 2080 + stamp. MPS 1500 C. Tutto a sole L. 890.000 -Telefonare ore pasti serali a Duracci Palma (x Tiziana). Tel. 081/8162622

Vendo C64 + Drive 1541 + Speeddos + Stampante MPS 802 + Eprom Grafica + Registratore C2N + 2 Joystick + adattatore telematico 6499 + programmi e istruzioni a L. 900.000. Maurizio Piraccini, Viale dei Pini 8 - Cesena. Tel. 0547/26958 (ore pasti).

Gruppo Comete 64 vende pacchi software ultime novità. Oltre 50 programmi. Es: CAD3D-CALC Result-Defender of the crown e tanti altri. Max serietà. Per informazioni: tel. 0833/782162 (Gianni).

Occasione: vendo Modem 6499, usato un mese, a L. 50.000 Gelmi Claudio, Via Brodolini 4 - 57023 Cecina (Li). Tel. 0586/680593

Vendo digilizzatore "Digi View" per Amiga, inclusi nel prezzo ottimi programmi per utilizzarlo. Il tutto a lire 80.000 (nella scatola ci sono anche i filtri per ottenere immagini a colori!). Istruzioni in italiano. Vendo inoltre campionatore di suoni per CBM 64, completo di microfono e/o software su disco e cassetta + cavi per collegarlo a impianti stereo ed altri. Questo Sampler è stato prodotto dalla Datel Electronics in Inghilterra. Il prezzo da me richiesto è di lire 80.000. Estrema facilità d'uso. Telefonare a: Paolo Sarego - 0746/484988. Annuncio sempre valido.

Cerco programmatore di Eprom per C=64. Artioli Marcello. Via Mario Agni 19 - 44100 Ferrara. Tel. 0532/54356

Cerco utenti Modem per collegamenti e scambi. Francesco Cannici, B.lo Q. Sella 18 - Novara. Tel. 0321/21447. P.S. (chiamate in 300/300).

Vendo per cambio sistema oltre 70 dischi con programmi per C64 - Joystick registratore - Copricomputer - Corso completo "Videobasic". Scrivere a Tonazzi Arturo, V. S. Giacomo 131 - 39050 S. Giacomo di Laives (Bz). Tel. 0471/940876

Per forzato inutilizzo vendo Commodore 128, registratore compatibile, Joystick, 300 programmi, un corso di basic su cassetta ed alcuni libri. Telefonare allo 0481/87631 (Gorizia).

Cerco Drive 1571 originale o compatibile max 2 anni di vita. Zona Catania e Provincia. Manuele Rocco, Via Vitt. Emanuele n. 124 - 95048 Scordia (Ct). Tel. 095/658352

Vendo modulatore video per Amiga 500 nuovo. Prezzo lire 38.000 trattabili. Possibilmente zona Asti. Gabriele Bellussi, Via Petrosi 27. Tel. 219206, prima delle 16.

Buy or swap the latest stuff from your favorite group WORLD OF WONDERS.
PLK. 037719 D - 5100 AACHEN 1 WEST GERMANY

Cerco in Parma amici 64isti amanti della programmazione (Basic.L.M.) per scambio informazioni e programmi. Telefonare ore pasti 0521/774407 chiedere di Luca. Annuncio sempre valido.

Vendo C16 in ottime condizioni + registratore + 3 giochi a L. 110.000. Vassalli Giuseppe. Tel. 02/347421. Dalle 19.00 alle 22.00.

Vendo giochi su disco X C64 a L. 6.000 il disco + spese spedizioni (pagamento in contrassegno) x richiedere la lista dei giochi scrivere a: Mastrantonio Giovanni, Via Dell'Impruneta 28 - 00146 Roma inviando L. 2.500 (in francobolli) x spese spedizioni.

Cerco Disck Drive 1541 per CBM 64 funzionante d'occasione. Solo Palermo e Provincia. Valdesi Giuseppe, Via Fondo La Manna 18 - 90135 Palermo. Tel. 091/551424

Vendo in blocco o separatamente dal n. 1 al n. 58 riviste (Commodore Computer Club) + n. 35 riviste (Elettronica 2000). Carmine: tel. 02/3533538 (ore 20.30).

Compro, vendo, scambio Software, accessori, manuali consigli e consulenze per Commodore. Rispondo a tutti. Marco Corazza, V. Machiavelli n. 3, Zola Predosa (Bologna). Tel. 051/751672

Vendo Commodore C64 + Disk Drive II 1541 + registratore + 250 programmi, SU.D e C il tutto 3 mesi di vita a L. 600.000. Telefonare a Luca: 039/463829

Cerco emulatore MS-DOS e GW-Basic a buon prezzo!!! (P.S. mi fido di voi!) Daniele Saguto, Via Nava, 12-95125 Catania. Tel. 095/336514

Vendo C64 + registratore + monitor + Modem 6499 + 2 Joysticks + 140 cassette (900 programmi) + corso "Parla-Basic" + riviste varie + scrivania porta computer a lire: 1.200.000. Per informazioni tel. 0371/65884 chiedere di Gianluca.

Cerco urgentissimamente Drive Ext. da 5 1/4 da collegare al Sidecar dell'Amiga 1000 (Drive Amiga 1020?). Roberto Anoardo, C.so Rosselli 44 - Torino. Tel. 011/3107569 (Ore ufficio).

Neo-utente Amiga "500", cerca software; disposto a trattare. Tel. (015)592459 ore serali. Spedite le vostre liste con modalità e recapito a: Pignataro Giuseppe, V. Repubblica n. 5 c.a.p. 13055 Occhieppo Inferiore (Vercelli). Risposte per tutti.

Vendo C128, drive 1570, MPS 1000 adattore telematico 6499, oltre 200 programmi solo zona Milano a L. 1.100.000, acquistabili anche separatamente. Tel. 02-4989830

Vega software vende eccezionali e nuovissimi videogames. (Tutti appena usciti!) per C=64. Antonio Tibaldo Via Martiri Res. 9 45023 Costa di Rovigo (Ro). Tel. 0425/97043. Annuncio sempre valido.

Vendo giochi stupendi per C=Amiga 500 (anche scambio). Per informazioni telefonare allo 0331/452901 - 20025 Legnano (Mi). Chiedi di Italo. Telefonare tra le ore 13.30-14.30.

Per C64 & Amiga scambio/offro ultime novità chiama dopo le 18.30 allo 095/430850 Catania o scrivi a: Torrisi Fabrizio, Via Vincenzo Giuffrida 107 Catania.

Vendo giochi ed utilies per C-16 e Plus/4 su cassetta e disco. A L. 3.000 l'uno. Per informazioni telefonare allo 080/435604. Chiedere di Fabio. Telefonare nel pomeriggio.

Vendo Commodore 128, drive 1541, registratore Datassette (nuovo), 3 Joystick, programmi su cassette e dischi (Print Master etc.) porta dischi con chiave a L. 600.000. Chiedere di Pietro ore pasti. Tel. 02/825.4353

Vendo C=64 con copritastiera, registratore, Disk Drive 1541, adattatore telematico, final cartige III, giochi in disco e cassetta, tutto L. 500.000. Tel. 02/2550705 - Massimo.

Vendo drive per CBM 64 compatibile + 65 dischi con programmi-portadischi 70 posti - The Tool Cartridge uguale alla Final Cartridge II - tutto in blocco per L. 250.000 oppure cambio con monitor ad ingresso R.G.B. Spera Giacomo Via Pretatti 50 - 67100 L'Aquila. Tel 0862/27055

Vendo C64, drive 1541 II, registratore originale, joystick professionale, rack di supporto, enciclopedia Jackson 500 giochi e utilies, light pen e riviste. Prezzo interessantissimo!! Chiedere di Daniele allo 040/821117

The Roman Importers/T.R.I. The best Amiga software in Italy! L. 2.000 for a disk! Silvio Picardo/-SPI Via B. degli Ubaldi 36 - 00167 Roma. Ph. 06/6235853

Vendo Commodore 128 D + interfaccia RS 232 C + programma originale G.E.O.S. V1.2/Lire 700.000. Tel. 010/383340

Vendo CBM64 New + registratore + 65 giochi. Fabio Misitano. Via G. Alberti 2. Catanzaro. Tel. 0961/20203

S.G.P.Z. Club. Compra-vende. Prog. X CBM 64 su disco o cass. S.G.P.Z. Via Zucchelli 16 Calderara di Reno (Bologna).

Cerco la "Niki-Cartridge". A buon prezzo e funzionante. Vendo tastiera "Music 64" 29 tasti cm. 75x16 con programma, funzionante. Tel. 0922/603469 (17-20) chiedere di Calogero.

Vendo fantastici giochi solo per Plus 4 Es: Sim.volo con sintesi vocale, calcio, scacchi 3 dimensioni e altri bellissimi G. 30. Scrivetemi e manderò la mia lista giochi. Balzano Giuseppe, Torre Annunziata - Via Oplonti 91 - 80058 Salerno.

Vendo giochi X C=64 al prezzo di L. 3.000. Telefonare allo 095/447496. Chiedere di Enzo. V. di Marco (Ct).

Amigasoft = A.P.G. Casella Postale 24027 Nembro (Bg)

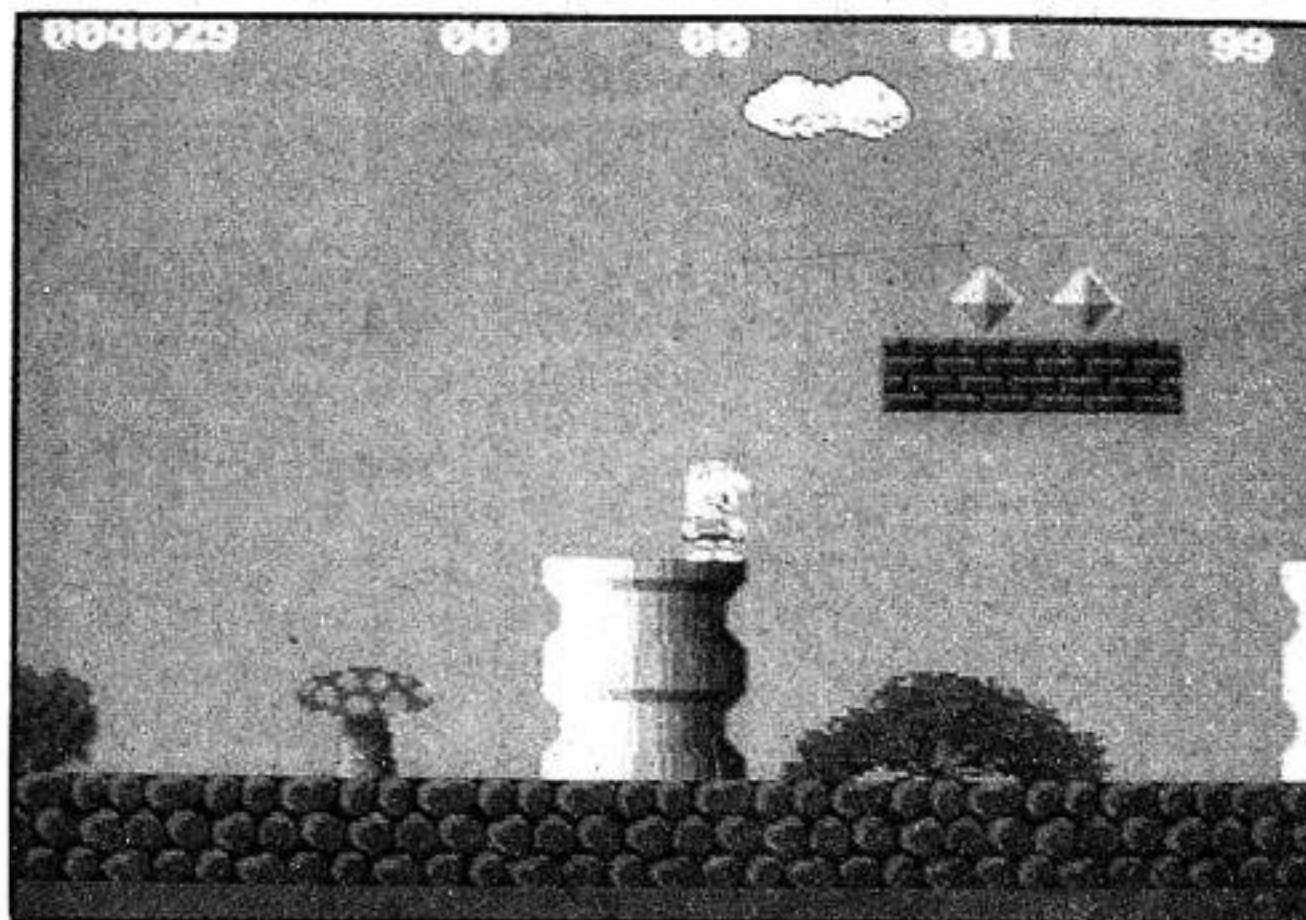
Cerco GW-Basic Emulater & MS-DOS per C=64 su cassetta. Contattate: Tardocchi Giuseppe, Via Kennedy 3 - 06019 Umbertide (Pg). Tel. 075/935025

Vendo giochi favolosi per C64, solo 3.000 a disco, entrambe le facciate. Ho tutte le novità. Simone Galigani. Via F.D. Guerrazzi n. 8/A. S.P. in Palazzi (Li). Telefono 0586/660070 tutte le ore.

Causa cambio sistema vendo miglior offerente tastiera C128+Drive 1541 + Stampante MPS803 + Joystick in possesso da 3 anni (eventuale software). Spedire Vs. offerta da min. L. 1.000.000 con recapito telefonico a Paolo Miro - Castello 3060/B (Ve).

Programmatore scambia-vende software x C64 in particolare sistemi per totocalcio basati su dati oggettivi con formule matematiche. Scrivere o telefonare a: Pasini Cristian - Via Fulda, 58 - 00148 Roma. Tel. 06/5234354

Scambio dischetti per Amiga, inoltre vendo C=16 + registratore + programmi a L. 100.000 trattabili. Scrivere a: Rosiello Angelo, Via Regina Elena 7, Sava (Ta). Tel. 099/6706572

Vendo registratore in ottime condizioni + 22 cassette con giochi x CBM 64 (circa 180 giochi) a L. 150.000! Claudio Portelli, P.za Bugiardini 2 - 63012 Cupra Marittima (A.P.). Tel. 0735/777756

VOLANDO SULLO SCHERMO

Una "banale" manipolazione dell'interrupt è alla base di scrolling puntiformi

di **Lorenzo Emilitri**

Uno degli effetti video più belli, tra quelli che lasciano senza fiato coloro che iniziano ad interessarsi di Raster, registri video e simili, è lo smooth scrolling verticale dello schermo, vale a dire lo spostamento, pixel per pixel, di zone più o meno vaste dello schermo.

Più volte, sia nei "demo" che circolano tra gli appassionati che in giochi commerciali, abbiamo visto schermate grafiche in multicolor "scivolare" senza alcuno sfarfallio verso la parte inferiore dello schermo visibile, lasciando all'utente l'inevitabile domanda: "Ma come diavolo hanno fatto?".

E poteva la rivista più amata dagli Italiani (C.C.C., naturalmente) non rispondere all'assillante quesito che lascia insonni milioni e milioni di smanettoni? (esagerare non guasta mai).

LA TEORIA...

Come molti di voi probabilmente sapranno, nel circuito video del C/64 (il famoso e pluricitato VIC-II) esiste un registro che si occupa dello smooth scrolling verticale.

Questa denominazione, forse oscura per i comuni mortali, significa che, suo tramite, è possibile spostare, sia pure di poco, il quadro video in senso verticale.

In altre parole i progettisti del C/64 hanno inserito il registro per permettere animazioni fluidissime anche in senso verticale. L'unica limitazione del registro è rappresentata dal fatto che non può spostare il video di più di sette pixel per volta, e che per realizzare animazioni di portata più ampia è necessario utilizzare altre tecniche software, più o meno simili a quelle che gestiscono lo scrolling orizzontale, argomento già trattato di recente su C.C.C.

La combinazione standard di hardware e software, pur essendo accettabile e nulla più quando si lavora su schermate di caratteri da far comparire e scomparire, dimostra tutta la propria inefficienza quando si lavora con schermate grafiche in alta risoluzione. In questi casi (sarà capitato anche a voi di esserne testimoni in qualche vecchio gioco, per esempio Bandit) il microprocessore non è in grado di spostare 8k di memoria in meno di 1/50 di secondo: l'effetto complessivo è molto scarso, con la schermata che si muove verso il basso a scatti.

Per risolvere il problema vi sono solo due possibili soluzioni (che ora, per la gioia di tutti gli studenti delle scuole superiori che detestano l'algebra) andremo a discutere:

A: Utilizzare un buffer di 8k in cui verrà ridisegnata la schermata, una parte ogni linea di scansione; la tecnica è abbastanza complicata e per ragioni di spazio non la descriveremo in questa sede. Come si può facilmente intuire, però, la soluzione è inaccettabile per tutti quei programmi (e giochi) che non si possono permettere di sprecare ben 8k di memoria da destinare ad un semplice buffer;

B: Adottare la tecnica descritta in questo articolo, che in sostanza consiste nell'imporre al VIC l'attesa di un numero variabile di linee prima di disegnare i caratteri. Ciò, come si può notare dal disassemblato, può essere svolto da una manciata di bytes in Linguaggio Macchina.

In effetti il programma presentato non fa altro che ingannare il VIC II, facendogli credere, tramite il registro di smooth scrolling, di dover aspettare per alcune linee prima di disegnare i caratteri.

Ciò è ottenuto semplicemente forzando nei 3 bit meno significativi del registro di smooth scrolling verticale il valore degli stessi bit del registro che rappresenta la posizione attuale del pennello elettronico.

Visto che il registro di smooth scrolling verticale è $D011, e che questo registro è VITALE per il corretto funzionamento del calcolatore, è chiaro che bisogna compiere questa operazione tramite AND e OR,

come riportato, del resto, nel disassemblato.

IL GIALLO DELL'ULTIMA LOCAZIONE

"Ma allora -direte voi- come mai sono state inserite parti che non hanno nulla a che vedere con questo?"

Quando diciamo che il VIC-II "aspetta" per alcune linee prima di disegnare i caratteri, commettiamo un'imprecisione: in effetti visualizza il contenuto dell'ultima locazione del banco video selezionato.

Senza appesantire eccessivamente la trattazione con una spiegazione completa dei banchi di memoria (per i più curiosi rimandiamo alla lettura dell'articolo "La gestione dei quattro banchi di memoria", pubblicato su C.C.C. numero 32), ci limiteremo a ricordare che il VIC "vede" solo 16K alla volta, e che è quindi costretto a dividere la memoria in quattro banchi separati (16 per 4 fa infatti 64. Elementare, mio caro Watson!).

Impostare a 0 l'ultima locazione del banco video selezionato è quindi l'unica soluzione per eliminare le fastidiosissime linee nere presenti quando si manda in esecuzione la routine.

Ma il banco video utilizzato all'accensione della macchina è il numero 3, ovvero quello che si estende dalla locazione 0 alla 16383. Scegliendo questo banco la memoria del BASIC dovrebbe essere limitata a 16382, dal momento che la locazione 16383 dovrebbe essere settata a 0. Perciò si è deciso di utilizzare per il video il banco 0, ovvero quello che si estende dalla locazione 49152 alla 65535.

Ciò non è così facile come si potrebbe pensare, visto che il VIC non è in grado di leggere la ROM dei caratteri che si trova a partire da $D000; ne consegue che la routine, facendo intenso uso della locazione 1, provvede a trasferire il set completo dei caratteri Commodore in RAM.

Ciò, pur significando un decisivo allungamento della parte in L/M, permette di avere a disposizione dell'utente TUTTA la memoria destinata al Basic, e scusate se è poco.

IL PROGRAMMA

Potete finalmente passare alla digitazione del programma che si limita a caricare in memoria, a partire da 49152, e lanciare la routine in Linguaggio Macchina, facendo apparire sul video una scritta che si muove su e giù per lo schermo.

La routine proposta è destinata ad essere estrapolata e modificata per adattarsi a vari programmi.

Per verificare se avete veramente appreso la tecnica descritta nell'articolo, vi consigliamo di realizzare uno smooth scrolling verticale di una schermata grafica multicolor (quelle salvate dai vari programmi di disegno artistico: Koala, Advanced Art Studio, Blazing Paddles eccetera).

Vi ricordiamo che, come si può notare dal disassemblato, impostando 0 come numero di linee da spostare, si ottiene la re-inizializzazione al valore standard del registro $D011.

Inoltre, per come è costruita la routine, essa "consuma" un numero di cicli-macchina per ogni quadro video direttamente proporzionale al numero di linee che volete spostare verso il basso. Ciò può risultare piuttosto gravoso per il processore, e portare ad un inaccettabile rallentamento i programmi Basic. Il suo uso è quindi sconsigliato in tutti quei casi in cui siano necessarie, ad esempio, complesse elaborazioni di tipo matematico o si debba gestire una gran quantità di dati.

E' invece tassativo disabilitarla qualora si debbano effettuare operazioni di I/O, quali la lettura di un file da disco o la stampa di un documento.

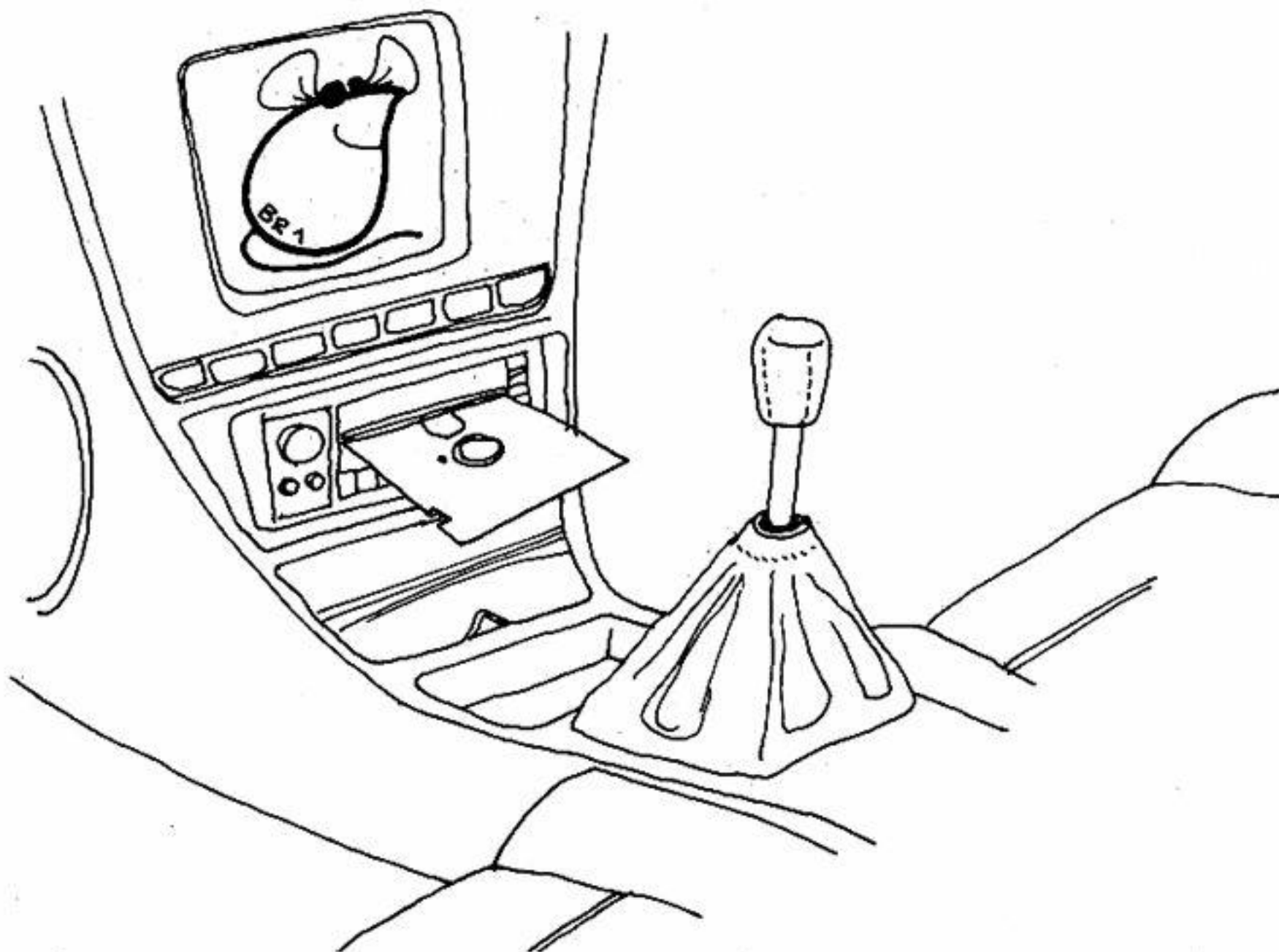
Per disabilitarla è sufficiente impartire...

POKE 53274, 0

...e, successivamente...

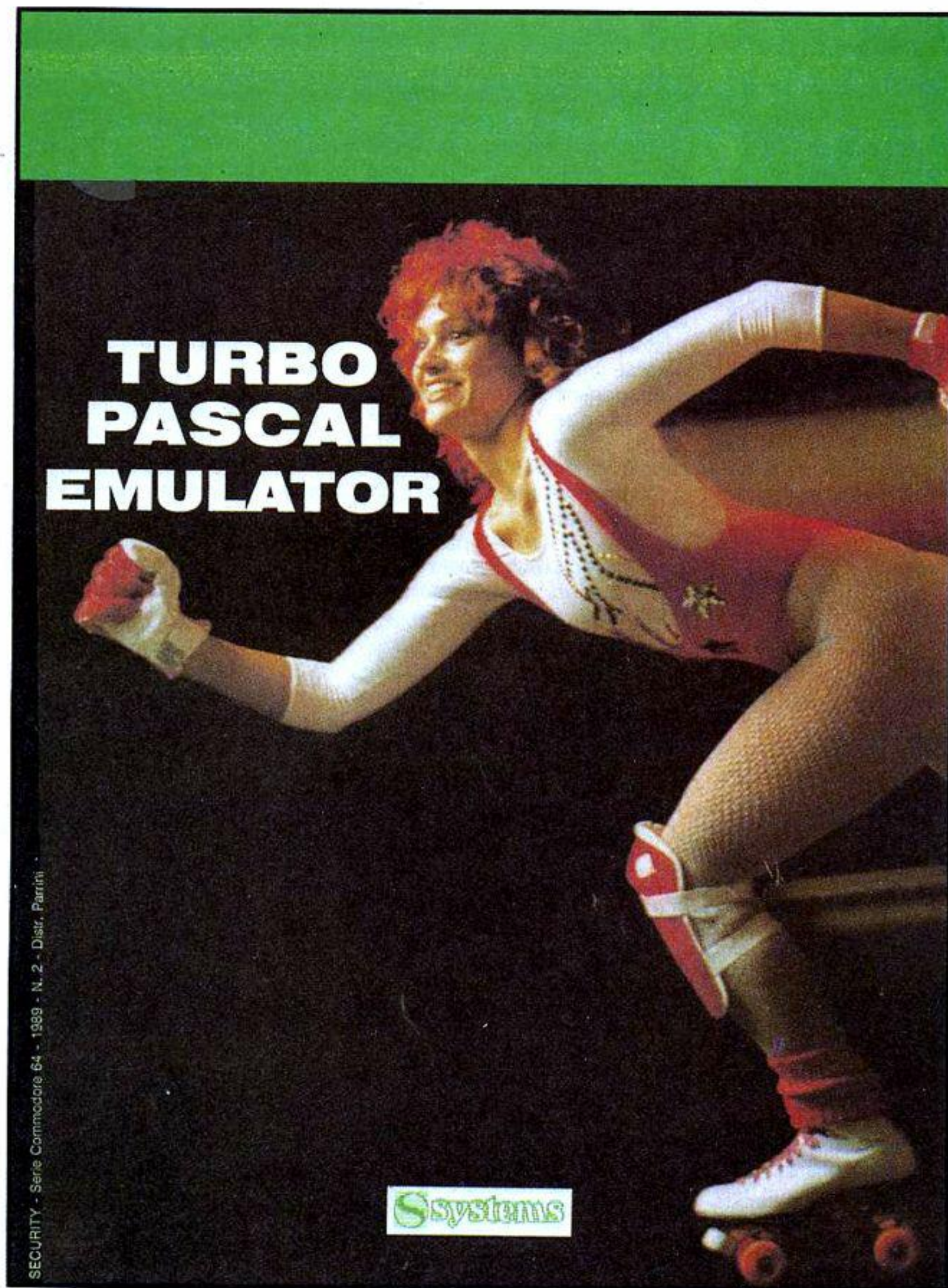
POKE 53274, 1

...per riabilitarla. Vi ricordiamo che mentre la routine è disabilitata non viene eseguita la scansione della tastiera e l'orologio interno (variabili TI e TI$) non viene aggiornato. E' quindi importante, prima di un'istruzione INPUT o GET, ricordarsi di riabilitarla.

```
100 REM *---------------------------*
105 REM *      MOVEDOWN V 2.00      *
110 REM *   BY L.EMILITRI - 1989    *
115 REM *---------------------------*
120 REM
125 REM * ALLOCAZIONE CODICI L/M *
130 FOR A = 49152 TO 49295
135 :   READ B : C=C+B
140 :   POKE A,B
145 NEXT A
150 IF C<>18297 THEN PRINT"ERRORE":END
155 SYS 49152 : PRINT CHR$(147)
160 REM * DEMO DEL PROGRAMMA *
165 POKE 53280,0 : POKE 53281,0
170 PRINT CHR$(158) TAB(12) "MOVEDOWN"
175 PRINT CHR$(30) TAB(11) "EMILITRI 89"
180 FOR A = 60 TO 120
185 :   FOR B = 0 TO 5
190 :   NEXT B
195 :   POKE 254,A
200 NEXT A
205 FOR A = 120 TO 60 STEP -1
210 :   FOR B = 0 TO 5
215 :   NEXT B
220 :   POKE 254,A
225 NEXT A
230 GOTO 180
235 STOP : STOP : STOP : STOP : STOP
240 REM
245 REM *   LINEE DATA PER L/M    *
250 REM *   VEDI DISASSEMBLATO    *
255 DATA 120,169,094,141,020,003,169
260 DATA 192,141,021,003,169,000,141
265 DATA 014,220,169,001,141,026,208
270 DATA 169,027,141,017,208,169,032
275 DATA 141,018,208,162,000,160,016
280 DATA 169,003,133,001,189,000,208
285 DATA 072,163,004,133,001,104,157
290 DATA 000,208,202,208,237,238,041
295 DATA 192,238,050,192,136,208,228
300 DATA 169,055,133,001,169,208,141
305 DATA 041,192,141,050,192,169,148
310 DATA 141,000,221,169,196,141,136
315 DATA 002,169,000,133,254,141,255
320 DATA 255,088,096,120,166,254,240
325 DATA 026,234,234,234,173,018,208
330 DATA 041,007,009,016,141,017,208
335 DATA 160,007,136,208,253,234,234
340 DATA 202,208,233,076,130,192,169
345 DATA 027,141,017,208,169,001,141
350 DATA 025,208,169,032,141,018,208
355 DATA 088,076,049,234
360 END
```

```
1  *------------------------------*
2  *        MOVEDOWN v 2.00        *
3  * BY LORENZO EMILITRI  - 040389 *
4  *------------------------------*
5                                  ;
6            ORG   $C000           ;INIZIO A 49152
7                                  ;
8            SEI                   ;DISAB. INTERRUPTS
9            LDA   #<IRQ           ;FA PUNTARE GLI
10           STA   $0314           ;INTERRUPTS
11           LDA   #>IRQ           ;ALLA LABEL 'IRQ'
12           STA   $0315           ;
13           LDA   #0              ;DISAB. TIMER
14           STA   TIMER           ;DEL CIA #1
15           LDA   #1              ;ABILITA I
16           STA   VICIRQ          ;RASTER INTERRUPTS
17           LDA   #$1B            ;LI FA PUNTARE
18           STA   HIRAST          ;ALLA RIGA $20
19           LDA   #32             ;
20           STA   RASTER          ;
21
22 * ---------------------------- *
23 * TRASFERISCE TUTTI I CARATTERI *
24 *     DA ROM A RAM  ($D000)     *
25 * ---------------------------- *

26           LDX   #$00            ;POS.: 0
27           LDY   #$10            ;N. PAGINE DA COPIARE
28 LP01      LDA   #3              ;ABILITA ROM
29           STA   $01             ;DEI CARATTERI
30 TR01      LDA   $D000,X         ;LEGGE DATO
31           PHA                   ;LO METTE NELLO STACK
32           LDA   #4              ;DISAB. CARATTERI
33           STA   $01             ;
34           PLA                   ;RIPRENDE VECCHIO DATO
35 TR02      STA   $D000,X         ;LO METTE A POSTO
36           DEX                   ;DECREMENTA CONTATORE
37           BNE   LP01            ;SE NON E' 0 VAI A LP1
38           INC   TR01+2          ;LEGGE E SCRIVE
39           INC   TR02+2          ;NELLA PAGINA SUCCESSIVA
40           DEY                   ;PAGINE FINITE?
41           BNE   LP01            ;NO, VAI A LP01
42           LDA   #$37            ;RIPRISTINA ROM
43           STA   $01             ;
44           LDA   #$D0            ;RIPRISTINA
45           STA   TR01+2          ;IND. D'INIZIO
46           STA   TR02+2          ;PER COPIA CARATTERI

47 * ---------------------------- *
48 * METTE A POSTO ALTRI PUNTATORI *
49 * VITALI DEL BASIC E DEL KERNEL *
50 * ---------------------------- *
51           LDA   #148            ;BANCO VIDEO
52           STA   VICBNK          ;NUMERO 0 ($C000-FFFF)
53           LDA   #196            ;PAGINA VIDEO
54           STA   PAGE            ;DA $C400
55           LDA   #0              ;DISABILITA LE
56           STA   $FE             ;RIGHE E
57           STA   LSTBNK          ;LO SPOSTAMENTO
58           CLI                   ;RIAB IRQ
59           RTS
60
61
62 * ---------------------------- *
63 * ROUTINE SOTTO RASTER INTERRUPT *
64 * ---------------------------- *
65 IRQ       SEI                   ;DISAB. IRQ
66           LDX   $FE             ;N. DI RIPETIZIONI
67           BEQ   EXIT01          ;SE NULLO VA A EXIT01
68 LP02      NOP                   ;BREVE
69           NOP                   ;RITARDO
70           NOP                   ;(6 CICLI)
71           LDA   $D012
72           AND   #$07            ;CALCOLI PER
73           ORA   #$10            ;IL VAL. DI $D011
74           STA   HIRAST          ;E IMMAGAZZINALO
75           LDY   #7              ;CICLO DI
76 LP03      DEY                   ;RITARDO
77           BNE   LP03            ;
78           NOP                   ;ALTRO RITARDO
79           NOP                   ;(4 CICLI)
80           DEX                   ;ALTRE RIPETIZIONI?
81           BNE   LP02
82           JMP   EXIT02          ;SI, VA' A EXIT02
83 EXIT01    LDA   #$1B            ;METTE A POSTO
84           STA   HIRAST          ;SHIFT VERTICALE
85 EXIT02    LDA   #1              ;CONFERMA ESECUZIONE
86           STA   VICFLG          ;DELL'INTERRUPT
87           LDA   #32             ;NUOVO INTERRUPT ALLA
88           STA   RASTER          ;RIGA 48
89           CLI                   ;RIAB. IRQ
90           JMP   NRMIRQ          ;ED ESCE
91 * ---------------------------- *
92 *        L A B E L S :          *
93 * ---------------------------- *
94 NRMIRQ    EQU   $EA31
95 VICIRQ    EQU   $D01A
96 VICFLG    EQU   $D019
97 RASTER    EQU   $D012
98 HIRAST    EQU   $D011
99 TIMER     EQU   $DC0E
100 LSTBNK   EQU   $FFFF
101 PAGE     EQU   648
102 VICBNK   EQU   $DD00
```

TURBO
PASCAL
EMULATOR

SECURITY - Serie Commodore 64 - 1989 - N. 2 - Distr. Parrini

systems

COME USARE L'EMULATORE "TURBO PASCAL 64"

Alcune precisazioni sul nuovo emulatore che si sta diffondendo a macchia d'olio in Italia

Annunciato sul N. 61 di C.C.C, il nuovo prodotto è presente da tempo in edicola, tanto che in molte rivendite è andato esaurito in poco tempo.

Grazie alla totale mancanza di protezione del disco, la possibilità di effettuare copie, soprattutto tra gli studenti delle scuole superiori, presenta alcuni problemi dal momento che, con la copia del disco, raramente circolano anche le fotocopie del manuale d'istruzione.

Alcuni dei lettori che hanno acquistato l'originale e coloro che posseggono una copia del nostro prodotto (ricordiamo, tuttavia, che è vietata la duplicazione a scopo di lucro), hanno lamentato difficoltà nella fase di compilazione dell'emulatore.

Siamo quindi giunti alla decisione di descrivere, nei minimi dettagli, le operazioni da compiere per compilare correttamente i programmi scritti in T-Pascal.

I FILE UTILI

Il dischetto originale (che deve esser subito duplicato ed accantonato in luogo sicuro) presenta la directory che vedete di queste pagine.

Dei vari file, quelli che devono esser presenti sul dischetto di lavoro sono i seguenti:

Turbo Pascal
Compiler
Linker
Library
Graph.p
Graph.bin

Non è quindi possibile operare se il dischetto di lavoro non contiene i file sopracitati. Gli altri file presenti nel dischetto originale sono dei banali separatori, oppure programmi di esempio (cerchi.pas, fatt.pas, lista.pas, hanoi.pas, grdemo.pas).

Pertanto copiate (con un qualsiasi copiatore di singoli file) su un dischetto TOTALMENTE VUOTO i sei file elencati prima e, per effettuare i nostri esperimenti, aggiungete SOLO il file "Cerchi.pas", che fungerà da cavia. La presenza di altri file potrebbe, almeno per il momento, creare problemi.

Caricate il primo file con un semplice...

Load "turbo pascal", 8

...e, al termine della brevissima operazione, digitate il consueto RUN.

Appare, a questo punto, la schermata iniziale del Turbo Pascal in cui sono evidenziati i sei comandi disponibili, attivabili grazie alla pressione del tasto corrispondente alle loro iniziali (Work file name, Edit, Compile, Run, Dir, Quit).

Esaminando la directory (premendo "D") dovrebbero comparire i sette nomi dei file presenti su disco. Premendo un tasto qualunque si ritorna invece al menu iniziale.

Premiamo, appunto, "W" per indicare al compilatore l'intenzione di caricare in memoria un file e, alla domanda che compare, rispondiamo digitando il nome dell'(unico) programma presente sul supporto magnetico: Cerchi.pas.

Se il nome non viene digitato ESATTAMENTE come risulta sulla directory, il compilatore provvederà automaticamente a crearne uno nuovo, dotato di tale nome.

Supponendo di aver digitato correttamente il nome "Cerchi.Pas", il drive ronzerà per alcuni secondi e ricomparirà la schermata iniziale.

Per sincerarvene, premete ora "E" (edit): il drive tornerà a ronzare per caricare il file richiesto. Quando compare su video il programma, non modificatelo in alcun modo (pena la generazione involontaria di qualche errore di sintassi), ma premete, in successione...

Ctrl K
Ctrl Q

...che ordinano al compilatore di ripresentare la schermata iniziale.

A questo punto possiamo passare alla compilazione vera e propria. Premete quindi "C".

Dopo un po' di tempo compare la domanda:

Do you want an exe file (y/n)?

alla quale risponderete affermativamente. Lo schermo scomparirà di nuovo per poi riapparire dopo alcuni secondi.

E' bene sottolinare che le varie operazioni impiegano un tempo direttamente proporzionale alla lunghezza (e complessità) del programma. Con un C/64 privo di qualsiasi velocizzatore e corredato di un normalissimo drive 1541, abbiamo rilevato, per il file "Cerchi.pas" i seguenti intervalli di tempo:

Caricamento: 6 sec.
Fase A: 2 min. 2 sec.
Fase B: 1 min. 43 sec.

Per "caricamento" intendiamo il tempo necessario a caricare il file dopo la pressione del tasto "E"; con "Fase A" ci riferiamo al tempo trascorso tra la pressione di "C" e la comparsa della domanda (Do you want...); con "Fase B", ovviamente, dalla pressione di "Y" alla ricomparsa del menu iniziale.

A questo punto tutto è in ordine: il programma è pronto per l'uso e risulta presente, compilato, su disco. Premendo "D"(ir), infatti, vedrete la presenza di un nuovo rigo...

65 "%cerchi.pas" prg

...che rappresenta, appunto, il programma compilato. Tornati al menu iniziale, e premendo "R", vedremo girare il programma, che ha lo scopo di generare vari cerchi concentrici.

In memoria è ora presente proprio il programma "%cerchi.pas". Per compilare un nuovo programma vi suggeriamo di spegnere il computer e caricare e lanciare nuovamente "Turbo Pascal" come, appunto, è riportato sul manuale d'uso.

Inutile ricordare che il programma "%cerchi.pas" (come qualsiasi altro programma compilato), può essere caricato e lanciato "da solo", anche in assenza, cioè, dei sei file del compilatore T. Pascal.

Per sincerarvene, spegnete il computer, riaccendetelo, digitate...

load "%cerchi.pas", 8

...estraete il dischetto dal drive e date il solito Run.

PRIMI GUAI

Alcuni accorgimenti sono però necessari per usare il compilatore senza avere guai di vario tipo.

Il primo riguarda la pazienza: avete letto bene! I tempi di compilazione possono essere piuttosto lunghi, ma usando opportuni velocizzatori SU CARTUCCIA (quelli software potrebbero sovrapporsi al compilatore e mandare tutto in Tilt) si possono anche dimezzare.

Tanto per fare alcuni esempi, ricordiamo che il prodotto funziona egregiamente anche con C/64 dotato di Speeddos, con C/128 (in modo, ovviamente, 64) e 1571, con C/64 dotato di cartuccia Robcom (da escludere mediante il tasto F7 dopo aver effettuato il caricamento del file "Turbo Pascal").

Altri velocizzatori hardware non dovrebbero presentare problemi. Vi consigliamo, tuttavia, di effettuare i primi esperimenti privando il vostro computer di qualsiasi cartuccia o procedura non standard. In seguito, presa confidenza con il compilatore, potrete verificare eventuali (in)compatibilità.

Un altro accorgimento (e il ricordarlo dovrebbe esser superfluo) riguarda il modo stesso di affrontare la stesura di un programma.

Un programma in Pascal, lo ricordiamo, non si può "improvvisare" alla tastiera come siete soliti fare, al contrario, con il Ba-

COME PROCURARSI T. PASCAL EMULATOR.

Coloro che non fossero riusciti a procurarsi in edicola l'Emulatore Turbo Pascal (che viene prodotto SOLO su disco e SOLO per C/64 o C/128 in modo 64) possono richiedere la confezione al nostro servizio arretrati.

Per maggiori informazioni si consiglia di telefonare solo al pomeriggio (dalle 14:30 alle 17:30) dei giorni feriali, sabato escluso. Tel. 02/84.67.348

sic. E' indispensabile studiare il problema da implementare, scriverlo con un word processor (Easy Script, o compatibile) e, dopo averlo esaminato con molta attenzione, effettuare i primi tentativi di compilazione. Eviterete, in tal modo, segnalazioni di errore che vi farebbero perdere il tempo che vi siete illusi di risparmiare scrivendo frettolosamente il programma.

Un altro inconveniente è rappresentato dagli stessi nomi dei file. Attenzione a non superare il numero dei caratteri consentiti e ricordate che il compilatore appone automaticamente il simbolo della percentuale (%) al file oggetto, incrementandone quindi la lunghezza.

Tentando di compilare un file il cui nome risulta già presente su disco, si genera un errore che costringerà il LED del drive a lampeggiare. Prima di compilare, accertatevi che non vi siano incompatibilità dovute ai nomi!

Se il listato sorgente (Ascii) non contiene errori, e si giunge quindi senza inconvenienti alla fatidica domanda "Do you want..." rispondete "Y" in modo da ottenere, su disco, il file compilato.

Questo consiglio è valido per due motivi. Il primo perchè potrete, in ogni caso, disporre di un programma oggetto completo e funzionante. Il secondo, poi, perchè in alcuni casi (rispondendo "N" alla domanda sulla creazione di un file exe) si può generare una condizione di errore su disco. Durante la fase di compilazione, infatti, vengono generati alcuni file sui quali il compilatore lavora e che verranno poi cancellati automaticamente al termine della procedura. Alcuni di questi, in casi particolari, possono influenzarsi negativamente in caso di mancata registrazione del file exe.

Se, quindi, non volete un file exe, provvedete a registare (Ctrl K, Ctrl S) la nuova versione del programma, in modo da esser sicuri di non perderlo nel caso sia necessario spegnere il computer per riprendere una nuova elaborazione.

In alcuni casi, se compare il messaggio "Read/write error or file not found", provate a registrare il file di cui non volete il file .exe, tornate al menu principale e caricatelo nuovamente ("W"). Se la procedura dovesse ancora fallire (caso piuttosto raro), spegnete il computer e riaccendetelo.

In quest'ultimo caso, prima di caricare nuovamente "T. Pascal", provvedete a duplicare il file che non ha generato errori di sorta con la solita sintassi...

```
Open 15, 8, 15, "C:nuovonome=vecchionome"
Close 15
```

...prima di ripetere tutto dall'inizio.

In altre parole, è probabile (ma, lo ripetiamo, raro) che, dopo la segnalazione di vari errori di sintassi ed elaborazioni più o meno intense, il compilatore perda... la bussola ed è necessario spegnere, riaccendere e riprendere il lavoro.

La segnalazione di inspiegabili "Read/write error or file not found" può esser quindi dovuta ad un fenomeno del tipo appena descritto, ma compare, soprattutto, dimenticando di registrare un programma digitato (o semplicemente corretto) servendosi di Edit.

Alla fine della digitazione (o della correzione di un file caricato da disco), infatti, deve tassativamente seguire la corrispondente registrazione con Ctrl K, Ctrl S. Il compilatore opera la compilazione sul file presente su disco e non su quello presente in memoria.

Se, quindi, dopo aver caricato un programma apportate modifiche e tentate di compilarlo (senza aver provveduto a registrare la nuova versione), verrà compilata la vecchia versione del programma presente su disco (in cui è ancora presente l'errore) e non quella nuova, contenente la correzione!

```
0   "tp emulator         " ax 2a
18     "turbo pascal"         prg
0      "----------------"     usr
115    "compiler"             prg
5      "linker"               prg
48     "library"              prg
0      "----------------"     usr
4      "graph.p"              seq
16     "graph.bin"            prg
0      "----------------"     usr
0.     "-     demo      -"    usr
0      "----------------"     usr
2      "cerchi.pas"           seq
2      "fatt.pas"             seq
6      "lista.pas"            seq
5      "hanoi.pas"            seq
2      "grdemo.pas"           seq
411  blocks free.
```

Pertanto dovete SEMPRE registare la nuova versione PRIMA di effettuare un tentativo di compilazione.

E' da tener presente che spesso, dopo che un programma è stato compilato e mandato in esecuzione ("R"), al termine di questo non si "rientra" in T. Pascal, soprattutto se il listato contiene istruzioni per l'alta risoluzione.

Inoltre è possibile che, dopo la compilazione, il tentativo di apportare modifiche al listato appena processato (inserimento nuove linee e così via) mandi in tilt il calcolatore. Niente paura! Il programma è ancora presente su disco: basterà spegnere, riaccendere e ricaricare il file (dopo averne, magari, effettuata una copia con altro nome).

Spesso, infine, nel caso di numerose segnalazioni di errore (in fase di compilazione) è sufficiente prestare attenzione solo alla prima di queste, dal momento che la sua presenza influenza la "logica" del listato rimanente. In un programma privo di errori, infatti, può succedere che l'introduzione di un solo errore (la cancellazione, ad esempio, di un punto e virgola) generi la segnalazione di altri errori, ad esso successivi, che in realtà non esistono.

Per sincerarvene, provate ad inserire volutamente un solo errore (magari all'inizio di un listato) in uno dei programmi dimostrativi presenti su dischetto e tentatene poi la compilazione.

CONCLUSIONI

Speriamo, con queste note, di aver eliminato i vari inconvenienti lamentati.

Ricordiamo, per finire, che nonostante sia presente un Editor di linea è bene seguire una procedura migliore, che vi consentirà di risparmiar tempo:

- scrivete con Easy Script il listato, prestando la massima attenzione a non commettere errori di alcun tipo (attenzione ai punti e virgola, vitali in T. Pascal!)
- caricate quest'ultimo file e tentate la compilazione.
- scrivete su carta gli eventuali errori visualizzati in fase di compilazione e studiate soprattutto il primo di essi.
- spegnete il computer, riaccendetelo, caricate e lanciate Easy Script.
- Individuate gli errori segnalati, provvedete a correggerli e registrate il nuovo file dotandolo, dato che ci siete, con un nuovo nome.
- ripetete la procedura dall'inizio, fino alla mancata segnalazione di errori.
- rispondere "Y" alla domanda "do you...".
- vivere felici e contenti.

Scherzi a parte, ci permettiamo di ricordare che siamo sempre a vostra disposizione per eventuali chiarimenti sul nostro prodotto.

A PROPOSITO DI Z-80

Alcuni (piccoli) esperimenti che dimostrano l'effettiva presenza, nel C/128, di un secondo microprocessore

di **Alessio Palma**

Tutti sanno che nel Commodore 128 è presente un altro processore, oltre all'8502; lo Z-80, infatti, garantisce il funzionamento dei programmi scritti in "ambiente" CP/M.

Quasi nessuno, però, conosce il sistema per attivare il secondo micro.

Prima di continuare il discorso, provate ad entrare in ambiente Monitor ed a scrivere (⟨ R ⟩ = premere il tasto Return):

D OFFDO OFFEO ⟨ R ⟩

Ciò che compare è il breve programma allocato nelle locazioni di memoria indicate nel comando; provate ora a lanciarlo con:

G OFFDO ⟨ R ⟩

L'effetto che ne consegue è la richiesta di inserire il disco del sistema CP/M. In altre parole, non avte fatto altro che attivare lo Z-80 che, a sua volta, richiede un "sistema" per l'inizio delle operarazioni.

Provate ora (magari dopo aver resettato il computer e riattivato il Monitor) a cancellare il programma con...

F OFFEO OFFDO OO ⟨ R ⟩

...e a controllare se è ancora presente:

D OFFEO OFFDO ⟨ R ⟩

Come potete notare il programma è sparito, proprio perchè era in una zona Ram.

Chi, quindi, lo aveva trascritto in quelle locazioni, e quando?

Prima di proseguire, esaminiamo ciò che combina il programma:

1) Disabilita le interruzioni.
2) Seleziona una particolare configuarazione di memoria.
3) Pone BO in D505... STOP! Avete letto bene: a questo punto della procedura viene attivato il micro Z-80 mentre l'8502 viene "bloccato"; tuttavia i suoi registri mantengono ancora il loro valore, come pure il Program Counter e così via.

Se ora provate a cancellare tutta la RAM O con...

F 1000 FFFF OO ⟨ R ⟩

...e poi a resettare, si nota che, oltre alla routine sopra citata, vengono scritti altri codici a partire da FFEO, proprio dopo la nostra routine.

I nuovi codici macchina, però (cosa molto interessante) non rappresentano alcuna istruzione nota all'8502; forse le ROM hanno qualche BUG, oppure trasferiscono, oltre a ciò che serve, anche un po' d'immondizia?

Le cose, ovviamente, non stanno così. Quando l'8502 viene "fermato", è necessario che un altro programma sia presente per attivare lo Z-80. Questo, dopo averlo eseguito, deve agire in modo analogo, "preparando" un programma che possa essere interpretato quando passerà la mano all'8502.

Il fatto che lo Z-80 debba eseguire un programma di autoesclusione, implica che

il programma stesso sia scritto in codice Z-80.

I codici del programma prima esaminato, privi di "senso" per l'8502, hanno invece un ben preciso significato per lo Z-80.

Una breve descrizione del modo di operare del programma è riportato, a parte, in queste stesse pagine.

Il fatto che l'ultima istruzione imponga un salto a 08 implica, da 08 in poi, la presenza di un programma l.m. scritto in codice Z-80; l'8502, però, non può proprio "vederlo" forse perchè è una rom riservata allo Z-80 e l'8502 non riesce ad accedervi, quasi come per la ram del chip video 8563.

Come si può notare nel disassemblato, all'indirizzo FFEB viene attivato l'8502 che continua a lavorare dal punto esatto in cui si era prima fermato. Ciò vuol dire che se riattiviamo lo Z-80 la prima istruzione ad essere eseguita sarà il NOP, poi il RST ed il mare di istruzioni idonee al caricamento del CP/M.

Che cosa può accadere se, al posto di NOP e RST 08, si pone un bel JP "inizio nostro programma"?

Succede che lo Z-80, semplicemente, lo esegue!

Attenzione, però: il programma in questione "deve" finire con JP FFE0, in modo da restituire il controllo all'8502 il quale si troverà, non appena verrà riattivato, un NOP e poi un JMP 1100...

Orrore! Il JMP 1100 esegue la routine di reset, cosa che, probabilmente, non vogliamo che accada.

Quindi, invece di JMP 1100, poniamo un CLI che riabilita le interruzioni dell'8502, seguito da RTS che ha il compito di terminare la routine.

RIASSUMENDO

Se vogliamo far eseguire allo Z-80 una nostra routine dobbiamo:

1) trascrivere nella locazione $1100 il codice CLI e nella $1101 il codice RTS; oppure eliminare JMP 1100 del programma di esclusione dell'8502.

2) Sostituire a NOP e RST 08 dello Z-80 un JP indirizzato verso la nostra routine.

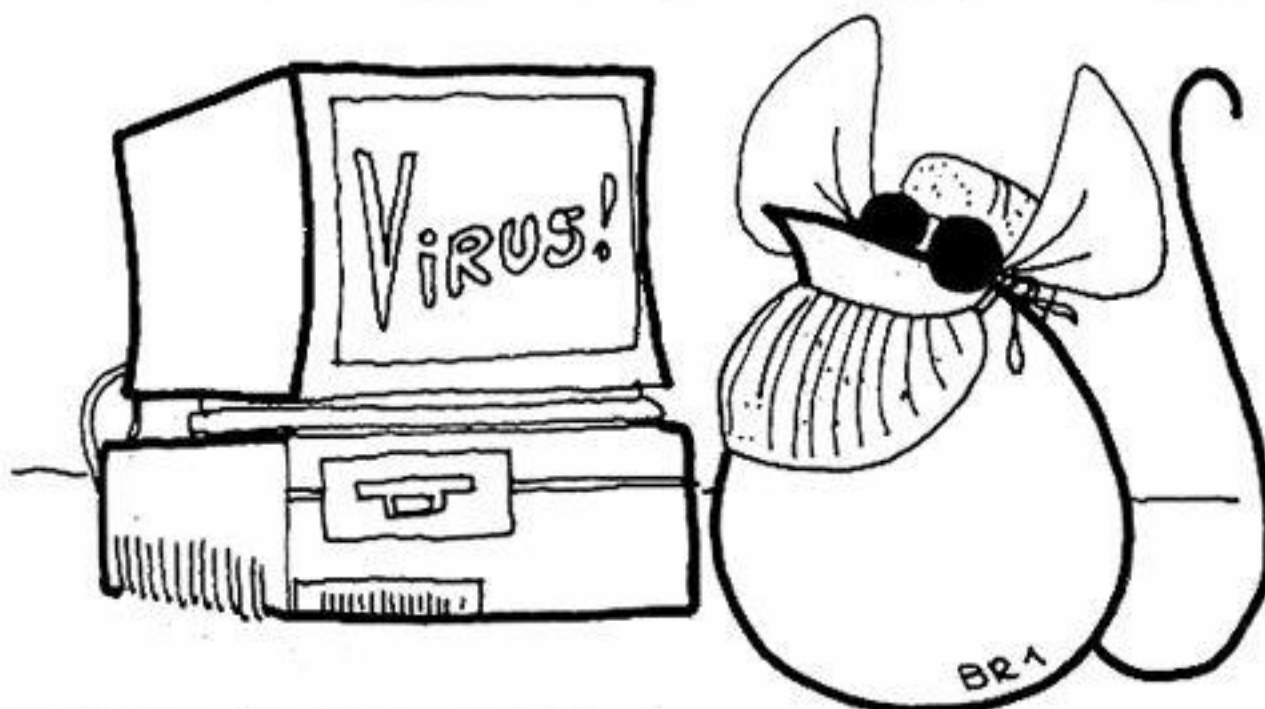
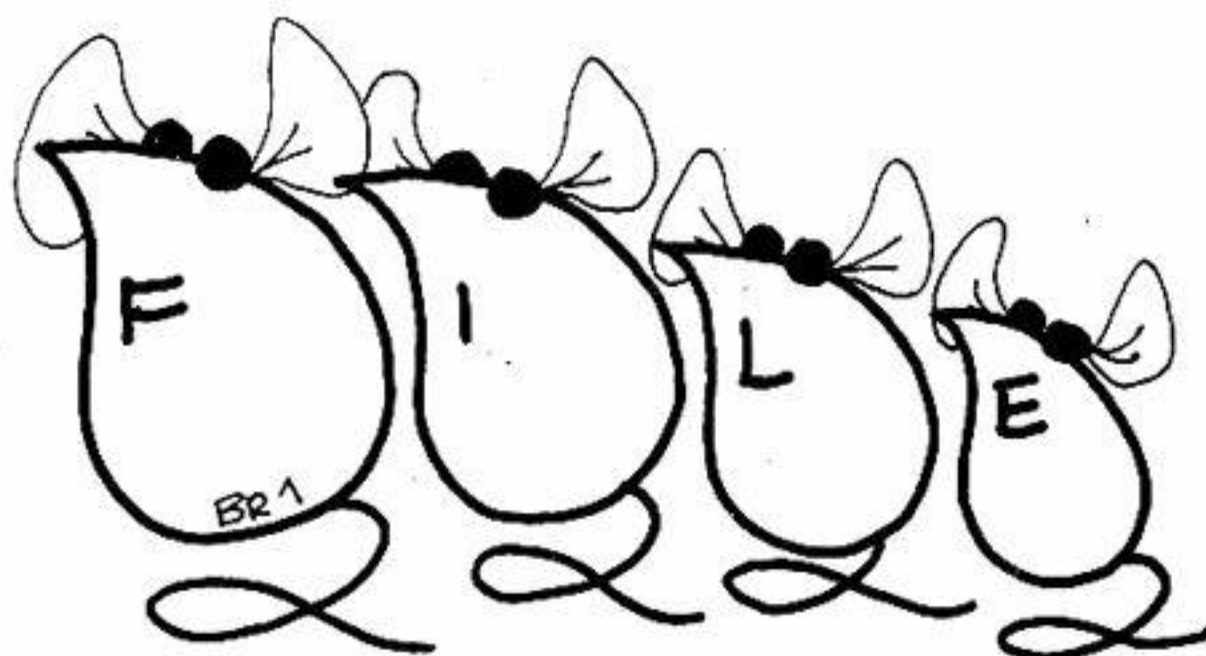
I due programmi di esempio saranno utilissimi a coloro che vogliono programmare in Z-80 anzichè 6502.

Ricordiamo che il set di istruzioni dello Z-80 è vastissimo e nettamente superiore a quello del 6502 e compatibili vari (8502, 7502, 65C02 ecc). Possiede, inoltre, registri a 16 bit con cui indirizzare la memoria ed eseguire operazioni aritmetiche.

Lavorare in C/128 (modo Z-80) può quindi essere una nuova occasione per conoscere le potenzialità di un microprocessore finora sconosciuto ai 128-isti. E' ovvio che l'impresa si rivela piuttosto ardua sia per la cronica carenza di informazioni sulla mappa di memoria dello sfortunato computer Commodore, sia perchè è necessario procurarsi un buon manuale sullo Z-80.

E andar per tentativi sperando in bene!

```
0FFE0 FE          DI            ;Disabilita' le interruzioni dello Z80
0FFE1 3E 3E       LD A,3E       ;Carica in accumulatore il valore 3E
0FFE3 32 00 FF    LD (FF00),A   ;Nuova configurazione di memoria
0FFE6 01 05 D5    LD BC,D505    ;Pone nel registro BC il valore D505
0FFE9 3E B1       LD A,B1       ;Carica in accumulatore il valore B1
0FFEB ED 79       OUT (C),A     ;Attiva l'8502 e si ferma.
0FFED 00          NOP           ;Non fa nulla.
0FFEE CF          RST 08        ;Salta a 08 e carica il CP/M
```

Disassemblato commentato della routine citata nell'articolo

```
100 REM PROGRAMMA N. 1 PER UTILIZZARE IL MICRO Z-80 SUL COMMODORE 128
110 REM QUESTO PROGRAMMA LANCIA IL PROGRAMMA BI BOOT DELLO Z-80
120 REM ALLOCATO DA 0000 A 1000.
130 :
140 PRINT"INSERISCI IL DISCO DEL CP/M E PREMI UN TASTO": GETKEY A$
160 BANK 0 : SYS DEC ("FFD0")
170 REM FA CONTINUARE LO Z-80; QUESTO "LANCIA" IL CP/M
180 END
```

```
100 REM  QUESTO PROGRAMMA (N.2) E'  UN ESEMPIO D'USO
110 REM  DEL MICRO Z-80. EFFETTUA IL REVERSE DELLA PAGINA
115 REM  GRAFICA (SOLO MODALITA': GRAPHIC 1 E 2)
120 REM
130 BANK 0:GRAPHIC1:GRAPHIC5
140 POKE DEC("FFEE"),DEC("C3")
150 POKE DEC("FFEF"),DEC("00")
160 POKE DEC("FFF0"),DEC("60")
170 REM  JP 2000H  SALTA AL PROGRAMMA POSTO A 2000
180 REM  QUESTO E' OTTENUTO MODIFICANDO IL RST 80 CHE FA PARTIRE IL CP/M
190 REM  IL QUALE SOTTO Z80 PUO' ESSERE FATTO PARTIRE CON UN SEMPLICE
200 REM  RET 80
210 POKE DEC("1100"),DEC("58"):REM CLI:RENDE SENSIBILE L'8502 ALLE INTERRUZIONI
220 POKE DEC("1101"),DEC("60"):REM RTS:FINE PROGRAMMA .. E RITORNA L BASIC
230 :
240 REM CARICATORE PROGRAMMA Z80.
250 I=DEC("6000"):REM INDIRIZZO INIZIO PROGRAMMA Z80.
260 DO
270 : READ A$               :REM CICLO LETTURA
280 : IF A$="*" THEN EXIT   :REM CODICI MACCHINA
290 : A=DEC(A$)             :REM IN CODICE
300 : POKE I,A              :REM Z-80
310 : I=I+1
320 LOOP UNTIL A$="*"       :REM ULTIMO DATO
330 GOTO 500
340 :
350 DATA21,00,20 :REM LD HL,2000
352 DATA7E       :REM LD A,(HL)
353 DATA2F       :REM CPL
354 DATA77       :REM LD (HL),A
355 DATA23       :REM INC HL
356 DATA3E,00    :REM LD A,00
357 DATABD       :REM CP L
358 DATA20,F7    :REM JR NZ,LOOP
359 DATA3E,40    :REM LD A,40
360 DATABC       :REM CP H
361 DATA20,F2    :REM JR NZ,LOOP
362 DATAC3,E0,FF :REM JP FFE0
363 DATA"*"
460 :
470 REM IL PROGRAMMA DEVE TERMINARE CON FFF0 PERCHE' DEVE RIDARE
480 REM IL CONTROLLO ALL'8502
490 REM
500 REM FA PARTIRE LO Z80...
510 BANK0:SYS DEC("FFD0")
520 REM
530 REM  (C) ALESSIO PALMA - PESCARA
```

LO SCHERMO RADDOPPIATO

Come ottenere 80 colonne sul video del vostro C/64

di **Gianluca Venturini**

Sul numero 56 di C.C.C. apparve un programma che permetteva di visualizzare ottanta colonne sullo schermo del C/16, utilizzando lettere di dimensioni dimezzate da "disegnare" sulla pagina grafica.

Ci siamo così proposti di realizzare un analogo programma per il C/64 ed il risultato finale altro non è che la fedele conversione della precedente routine L.M. per C/16, come si potrà anche vedere confrontando i disassemblati delle due versioni.

Di rigore è l'utilizzo di un monitor (meglio se monocromatico) per distinguere i caratteri (poco chiari a causa delle dimensioni ridotte di 4 x 8 pixel) e non affaticare la vista.

LA ROUTINE L.M.

E' doveroso dire che la routine non permette di usufruire di un sistema operante ad ottanta colonne effettive così come normalmente si edita con le normali quaranta. La routine, infatti, si limita (ma non è poco!) a scrivere sulla pagina grafica i soliti caratteri, ma con dimensioni dimezzate, senza che ci dobbiamo preoccupare di effettuare lenti plottaggi o calcoli di coordinate.

Prima di "scrivere" in modalità 80 colonne è necessario impartire una sys (che verrà comunicata dallo stesso programma

caricatore e che, nel listato di queste pagine, inizia da 49434) che attiverà, a sua volta, una seconda routine (il cui disassemblato commentato è anch'esso presente) che ricopierà il set di caratteri allocato in ROM a partire da 57344 nella sottostante RAM per consentire un accesso semplificato alla routine di scrittura.

Per attivare la routine vera e propria è invece necessario digitare...

SYS in, ri, co, stringa, modo

...in cui:
"in" è di solito 49152 (ma, in caso contrario, verrà comunicato dal programma caricatore).
"ri" deve esser compreso tra 0 e 24 ed indica la riga video.
"co" è compreso tra 0 e 79 e si riferisce alla colonna video "virtuale" a partire dalla quale stampare la stringa; questa può essere una variabile stringa o un messaggio racchiuso tra apici.
"mode" può assumere i valori da 0 a 3, assumendo i significati che seguono:
mode = 0 maiuscolo normale.
mode = 1 maiuscolo reverse.
mode = 2 minuscolo normale.
mode = 3 minuscolo reverse.

Per comodità è inserita anche una routine di cancellazione pagina grafica (banale) allocata normalmente a partire da 49481.

Ultimo, ma non meno importante, è da segnalare il particolare che le routine di queste pagine sono interamente RILOCABILI e si distinguono, pertanto, da altre routine di pubblico dominio (per non parlare della possibilità di modificarle a piacimento grazie al prezioso aiuto fornito dal disassemblato commentato).

IL PROGRAMMA CARICATORE

Il listato Basic, che vedete pubblicato, serve, ovviamente, per caricare in memoria i dati relativi alle routine in linguaggio macchina; una volta completato il compito vengono visualizzati alcuni messaggi di utilità scritti sullo schermo ad ottanta colonne.

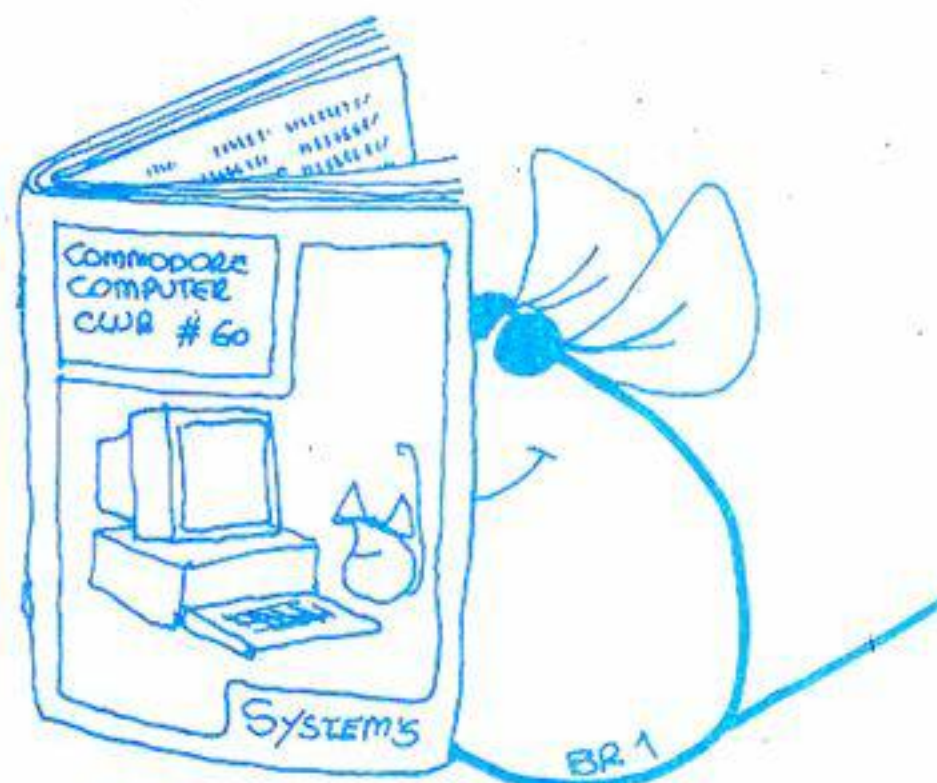
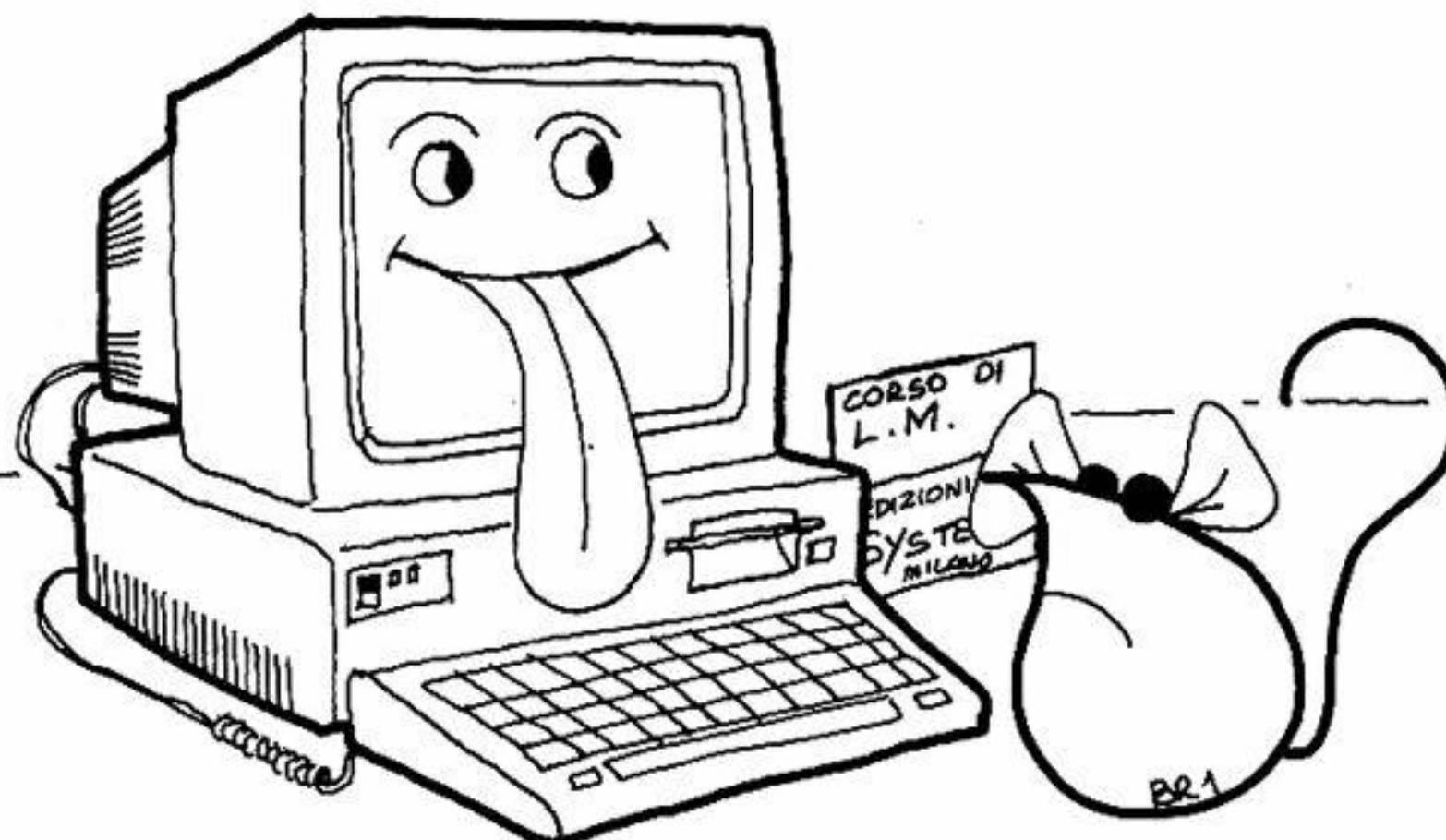
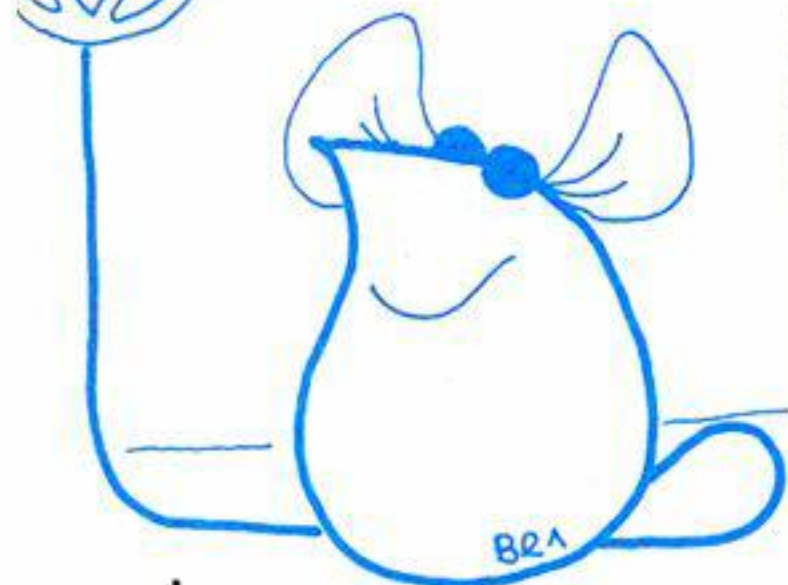
La pagina grafica su cui la routine scrive è quella posta nella RAM al di sotto del KERNAL, che si seleziona con i comandi...

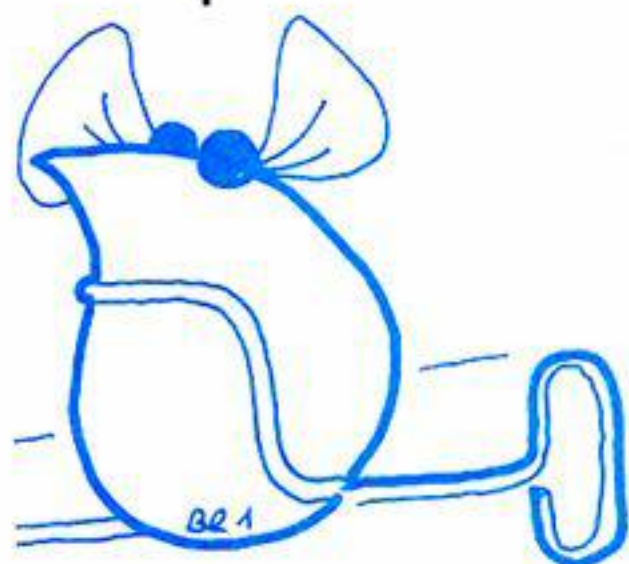
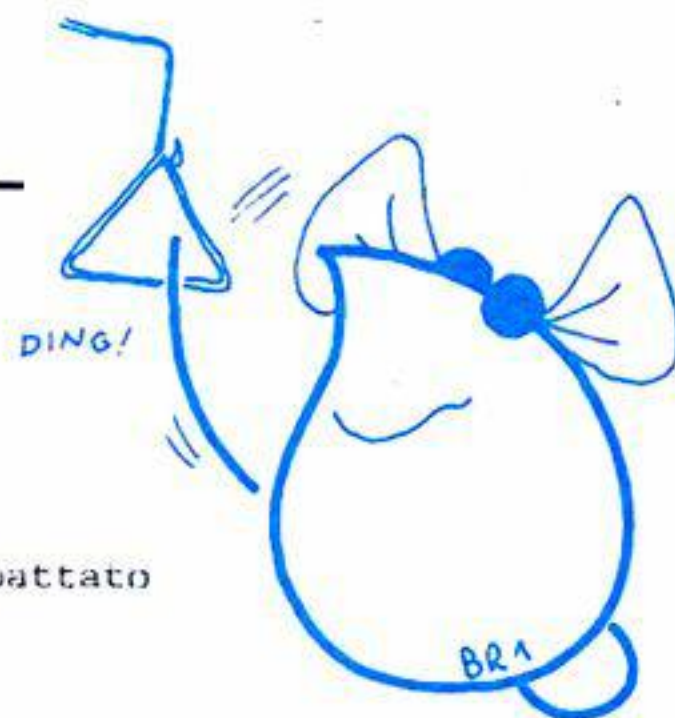
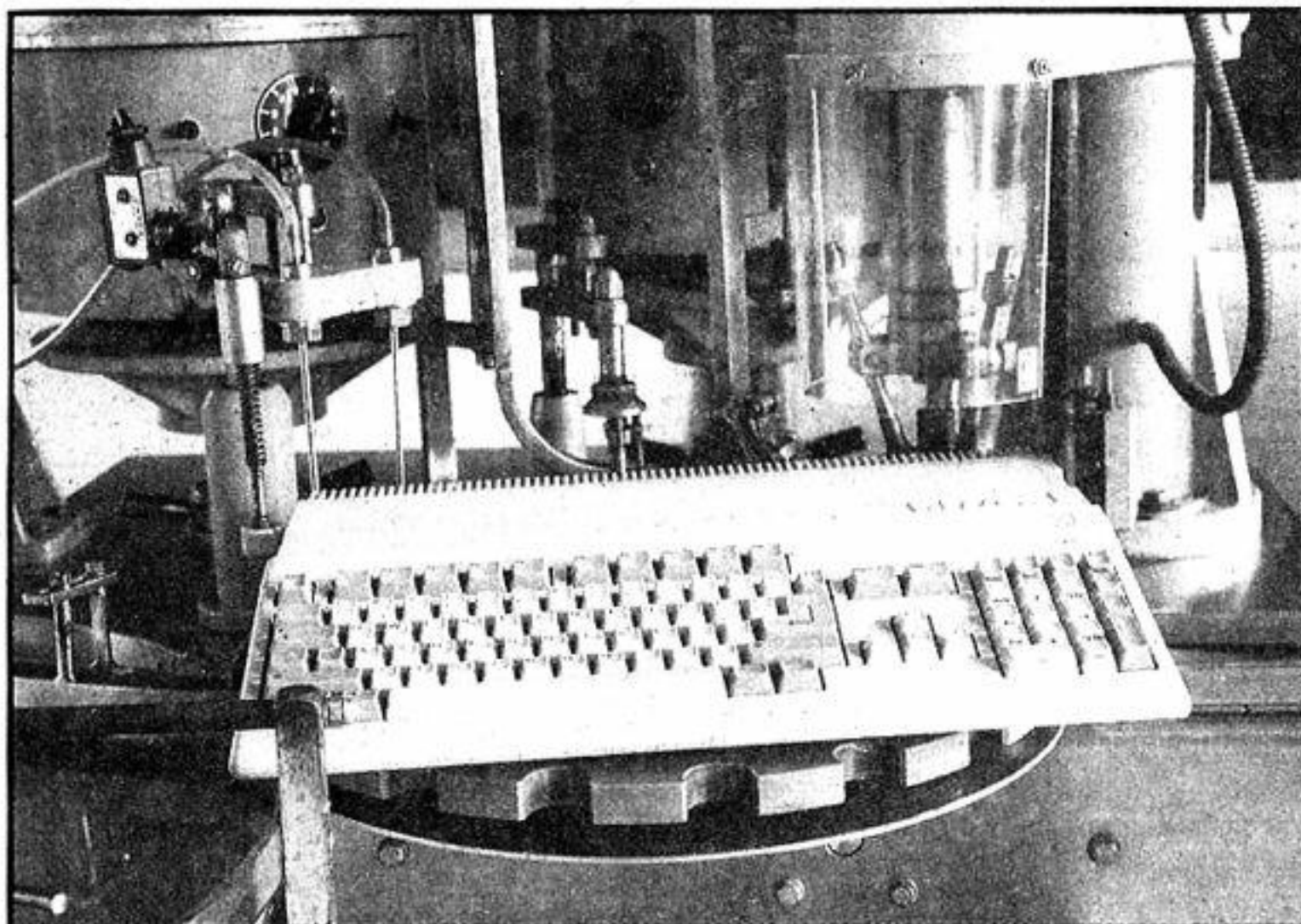
Poke 53272, 56: Poke 53265, 59: Poke 56576, 148

...che ormai dovrebbero(!) essere noti a tutti. La memoria colore per la stessa pagina grafica è posta da 52224 a 53223; per selezionare i colori e lo sfondo valgono gli stessi criteri già ampiamente discussi su C.C.C. a proposito di alta risoluzione, multicolor, e così via... (se proprio non lo sapete il valore è:
colore sfondo + colore inchiostro * 16

Il listato è comunque sufficientemente commentato per ciò che riguarda le norme di uso della routine.

```
10 REM 80 COLONNE SUL C/64
20 REM
100 DATA 169,208,133,251,32,253,174,32
101 DATA 158,183,224,25,176,12,134,140
102 DATA 32,253,174,32,158,183,224,80
103 DATA 144,5,162,14,76,139,227,134
104 DATA 139,32,253,174,32,158,173,32
105 DATA 130,183,165,34,133,178,165,35
106 DATA 133,179,132,141,32,253,174,32
107 DATA 158,183,224,4,176,220,138,41
108 DATA 1,133,254,138,41,2,240,4
109 DATA 169,216,133,251,120,169,52,133
110 DATA 1,169,0,133,142,169,224,24
111 DATA 101,140,133,143,165,140,74,102
112 DATA 142,74,102,142,24,101,143,133
113 DATA 143,165,139,74,10,10,10,144
114 DATA 2,230,143,24,101,142,133,142
115 DATA 144,2,230,143,160,0,132,253
116 DATA 234,177,178,16,16,41,127,201
117 DATA 127,208,2,169,94,201,32,144
118 DATA 6,5,64,208,14,201,32,144
119 DATA 123,201,96,144,4,41,223,208
120 DATA 2,41,63,162,0,134,4,10
121 DATA 38,4,10,38,4,10,38,4
122 DATA 133,3,165,4,5,251,133,4
123 DATA 44,208,195,160,7,162,3,169
124 DATA 0,133,252,177,3,74,74,102
125 DATA 252,202,16,249,165,139,41,1
126 DATA 240,39,165,252,74,74,74,74
127 DATA 133,252,177,142,166,254,240,3
```

```
128 DATA 9,15,44,41,240,69,252,145
129 DATA 142,136,16,209,169,8,24,101
130 DATA 142,133,142,144,22,230,143,208
131 DATA 18,177,142,166,254,240,3,9
132 DATA 240,44,41,15,69,252,145,142
133 DATA 136,16,178,230,139,164,253,200
134 DATA 196,141,208,165,169,55,133,1
135 DATA 88,96,169,0,133,3,169,208
136 DATA 133,4,120,160,0,162,51,134
137 DATA 1,174,0,208,177,3,162,52
138 DATA 134,1,174,0,208,145,3,200
139 DATA 208,235,230,4,165,4,201,224
140 DATA 208,225,169,55,133,1,88,96
141 DATA 169,0,133,3,169,224,133,4
142 DATA 162,32,160,0,169,0,145,3
143 DATA 200,208,249,230,4,202,208,242
144 DATA 96,41946
150 IN=49152:CK=0:PRINT"ATTENDI..."
155 FORX=0TO352:READA:POKEIN+X,A:CK=CK+A:NEX1
160 READCC:IFCK<>CCTHENPRINT"ERRORE NEI DATA":END
197 :
198 REM -- INIZIO PROGRAMMA --
199 :
200 FORI=0TO999:POKE52224+I,1:NEXT
205 REM DA 52224 A 53223 MEMORIA COLORE PER LA PAGINA GRAFICA
209 :
210 SYSIN+282:REM COPIA IL SET CARATTERI (NECESSARIO !)
219 :
220 SYSIN+328:REM  PULISCE PAGINA GRAFICA
229 :
230 POKE53272,56:POKE53265,59:POKE56576,148
231 REM VISUALIZZA LA PAGINA GRAFICA POSTA SOTTO IL KERNAL (57344-65535)
239 :
240 SYSIN,1,1,"80 COLONNE VERSIONE PER C-64 ",1
245 SYSIN,1,31,"PROGRAMMA SCRITTO DA G",0
250 SYSIN,1,53,"IANLUCA ",2:SYSIN,1,61,"V",0:SYSIN,1,62,"ENTURINI",2
260 SYSIN,3,1,"UTILIZZO:",2
265 A$="SYS"+STR$(IN)+",RIGA,COLONNA,STRINGA,MODO":SYSIN,3,11,A$,0
270 SYSIN,5,11,"RIGA,COLONNA:",0
275 SYSIN,5,25,"COORDINATE DA CUI STAMPARE CIO' CHE SI DESIDERA;",2
280 SYSIN,7,11,"      STRINGA:",0
285 SYSIN,7,25,"MESSAGGIO CHE SI VUOLE STAMPARE; PUO' ESSERE UNA",2
290 SYSIN,8,25,"VARIABILE STRINGA O UN MESSAGGIO RACCHIUSO TRA APICI;",2
295 SYSIN,10,11,"         MODO:",0
300 SYSIN,10,25,"PARAMETRO CHE PUO' ASSUMERE I SEGUENTI VALORI:",2
305 SYSIN,12,34,"0 : PER SCRIVERE IN MAIUSCOLO NORMALE",0
310 SYSIN,13,34,"1 : ",0:SYSIN,13,38,"PER SCRIVERE IN MAIUSCOLO REVERSE",1
315 SYSIN,14,34,"2 : ",0:SYSIN,14,38,"PER SCRIVERE IN MINUSCOLO NORMALE",2
320 SYSIN,15,34,"3 : ",0:SYSIN,15,38,"PER SCRIVERE IN MINUSCOLO REVERSE",3
325 SYSIN,20,1,"PREMI UN TASTO ",0
330 FORI=0TO60:GETA$:IFA$<>""THEN350
335 NEXT:SYSIN,20,1,"PREMI UN TASTO ",1
340 FORI=0TO60:GETA$:IFA$<>""THEN350
345 NEXT:GOTO325
349 :
350 POKE53265,27:POKE53272,21:POKE56576,151:REM RITORNO CONDIZIONI NORMALI
360 PRINT"♥BUON LAVORO":END
```

```
 1 ********************************
 2 * 80 COLONNE versione per C=64
 3 * scritto da Gianluca Venturini
 4 *    (Programma  RILOCABILE!)
 5 *    Disassemblato commentato
 6 ********************************
 7 INIZIO    LDA   #$D0     ; carica valore set caratteri maiuscolo
 8           STA   $FB
 9           JSR   $AEF0    ; prende una virgola
10           JSR   $B79E    ; e un dato a 8 bit in X  (RIGA)
11           CPX   #$19     ; se maggiore di 24
12           BCS   ERRORE   ; ERRORE
13           STX   $BC      ; altrimenti lo salva
14           JSR   $AEFD    ; preleva virgola
15           JSR   $B79E    ; e parametro COLONNA
16           CPX   #$50     ; che deve essere
17           BCC   CONT     ; compreso tra 0 e 79
18 ERRORE    LDX   #$0E     ; stampa
19           JMP   $E38B    ; ILLEGAL QUANTITY ERROR ed esce.
20 CONT      STX   $BB      ; salva COLONNA
21           JSR   $AEFD    ; preleva virgola
22           JSR   $AD9E    ; e una
23           JSR   $B782    ; stringa
24           LDA   $22      ; byte low puntatore alla stringa
25           STA   $B2      ; in $B2
26           LDA   $23      ; byte hi puntatore alla stringa
27           STA   $B3      ; in $B3
28           STY   $BD      ; e lunghezza in $BD
29           JSR   $AEFD    ; preleva virgola
30           JSR   $B79E    ; e parametro MODE
31           CPX   #$04     ; compreso tra 0 e 3
32           BCS   ERRORE   ; altrimenti errore
33           TXA            ;
34           AND   #$01     ; flag di reverse
35           STA   $FE      ; in $FE
36           TXA
37           AND   #$02     ; se maiuscolo
38           BEQ   CONT9    ; continua
39           LDA   #$D8     ; altrimenti memorizza
40           STA   $FB      ; valore set caratteri minuscolo
41 CONT9     SEI            ; disabilita gli interrupts
42           LDA   #$34     ; attiva
43           STA   $01      ; la RAM 'nascosta'
44           LDA   #$00     ; carica
45           STA   $8E      ; l'indirizzo
46           LDA   #$E0     ; della pagina grafica
47           CLC            ;   e
48           ADC   $8C      ; lo somma
49           STA   $8       ; F
50           LDA   $8C      ;   al
51           LSR   A        ; numero
52           ROR   $8E      ;
53           LSR   A        ;  di
54           ROR   $8E      ; riga
55           CLC            ;
```

```
56          ADC   $8F       ; moltiplicato per 320
57          STA   $8F       ; quindi salva il valore
58          LDA   $8B       ; carica numero colonna
59          LSR   A         ;
60          ASL   A         ;
61          ASL   A         ; lo moltiplica
62          ASL   A         ;
63          BCC   CONT2     ; per 4
64          INC   $8F       ;
65 CONT2    CLC             ;
66          ADC   $8E       ;
67          STA   $8E       ; e lo somma
68          BCC   CONT3     ;
69          INC   $8F       ; in $8E , $8F
70 CONT3    LDY   #$00      ; azzera contacaratteri stringa
71          STY   $FD       ;
72          NOP             ;
73          LDA   ($B2),Y   ; preleva un carattere dalla stringa
74          BPL   CONT5     ;
75          AND   #$7F      ; routine
76          CMP   #$7F      ;
77          BNE   CONT4     ;    di
78          LDA   #$5E      ;
79 CONT4    CMP   #$20      ; conversione
80          BCC   CONT6     ;
81          ORA   #$40      ; carattere
82          BNE   CONT7     ;
83 CONT5    CMP   #$20      ;    da
84 CONT6    BCC   RETURN    ;
85          CMP   #$60      ; ascii
86          BCC   CONT8     ;
87          AND   #$DF      ;    a
88          BNE   CONT7     ;
89 CONT8    AND   #$3F      ; codice schermo
90 CONT7    LDX   #$00      ; routine
91          STX   $04       ;   di
92          ASL   A         ; calcolo
93          ROL   $04       ;  della
94          ASL   A         ; posizione
95          ROL   $04       ;  del
96          ASL   A         ; carattere
97          ROL   $04       ; nella
98          STA   $03       ; tabella
99          LDA   $04       ; caratteri
100         ORA   $FB       ; con inizio $D000 per set maiuscolo
101         STA   $04       ; e $D800 per set minuscolo
102 SALTOO  BIT   $C300     ; maschera di salto
103         LDY   #$07      ; routine
104 CARAT   LDX   #$03      ;   di
105         LDA   #$00      ; compattamento
106         STA   $FC       ;    del
107         LDA   ($03),Y   ;
108 CHAR    LSR   A         ; carattere
109         LSR   A         ;
110         ROR   $FC       ; trovato
111         DEX             ;
112         BPL   CHAR      ; il carattere compattato e' in $FC
113         LDA   $8B       ; carica numero colonna
114         AND   #$01      ; controlla se e' pari o dispari
115         BEQ   PARI      ; salta a stampa n. pari
116         LDA   $FC       ; carica carattere compattato
117         LSR   A         ;
118         LSR   A         ;
119         LSR   A         ;
120         LSR   A         ; trasla tutto a destra
121         STA   $FC       ; e salva
```

```
122          LDA  ($8E),Y  ; preleva attuale carattere
123          LDX  $FC      ; controlla reverse
124          BEQ  JUMP0+1  ;
125          ORA  $0F      ; esegui reverse
126 JUMP0    BIT  $F029    ;
127          EOR  $FC      ; somma carattere trovato con compattato
128          STA  ($8E),Y  ; memorizza in attuale posizione
129          DEY           ;
130          BPL  CARAT    ;
131          LDA  #$08     ; routine
132          CLC           ;   di
133          ADC  $8E      ; incremento
134          STA  $8E      ;   della
135          BCC  SALTO1   ; posizione
136          INC  $8F      ;    del
137          BNE  SALTO1   ; carattere in HiRes
138 PARI     LDA  ($8E),Y  . stampa del carattere pari
139          LDX  $FE      ; controllo reverse
140          BEQ  JUMP1+1  ;
141          ORA  $F0      ; esegui reverse
142 JUMP1    BIT  $0F29    ;
143          EOR  $FC      ;
144          STA  ($8E),Y  ; stampa carattere compattato
145          DEY           ;
146          BPL  CARAT    ;
147 SALTO1   INC  $88      ; incrementa numero colonna
148          LDY  $FD      ;
149          INY           ; incremento puntatore al carattere stringa
150          CPY  $8D      ; controlla se fine stringa
151          BNE  SALTO0+1 ; se no allora contrinua
152 RETURN   LDA  #$37     ; altrimenti
153          STA  $01      ; riabilita interprete BASIC
154          CLI           ; e interrupts
155          RTS           ; ed esci.

156 ********************************
157 * Seconda routine :
158 * ricopia il set caratteri
159 * contenuto nella ROM da 57344
160 * nella sottostante zona
161 * per consentirne un facile
162 * accesso in ambiente HiRes
163 ********************************
164 ROUTINE2 LDA   #$00     ; inizio
165          STA   $03      ; sovrascrittura.
166          LDA   #$D0     ; dalla zona
167          STA   $04      ; RAM da $D000.
168          SEI            ; disabilita interrupts.
169 QUI0     LDY   #$00     ; pone Y=0.
170 QUI1     LDX   #$33     ; attiva la
171          STX   $01      ; ROM caratteri 'nascosta'.
172          LDX   $D000    ; 'accende la ROM'.
173          LDA   ($03),Y  ; preleva un dato.
174          LDX   #$34     ; seleziona la
175          STX   $01      ; RAM sotto la ROM caratteri
176          LDX   $D000    ; la 'accende'
177          STA   ($03),Y  ; e copia il dato.
178          INY            ; ripete
179          BNE   QUI1     ; finche' Y=0.
180          INC   $04      ; incrementa pagina.
181          CMP   #$E0     ; fino a ricopiare
182          BNE   QUI0     ; tutto il set caratteri.
183          LDA   #$37     ; riabilita
184          STA   $01      ; l' interprete BASIC,
185          CLI            ; gli interrupts
186          RTS            ; e ritorna al BASIC.
```

LA DIRECTORY A PORTATA DI MANO

Un modo semplice, in Basic, per archiviare selettivamente il contenuto dei nostri dischetti

Gli Amici di Amiga

La crescente diffusione di Amiga, sempre più competitivo anche a livello di prezzo, ha dato il via agli stessi fenomeni avveratisi per l'ex rivoluzionario C/64.

Ci riferiamo, in particolare, all'interesse delle software house per questo computer, con relativa abbondanza di produzione ad esso dedicata, ma anche, se non soprattutto, alla miriade di interscambi tra utenti.

La tecnica del "io do un programma a te se tu ne dai uno a me", carenza di manuali a parte, in genere fa sì che ognuno disponga di ingenti quantitativi di software, magari per lo più... inesplorati.

Un dischetto di Amiga, inoltre, anche quando sembra contenere un solo programma, spesso riserva gradite sorprese per i più smanettoni.

Basta "scavare" un po' nelle sue subdirectory (soprattutto la C), ed ecco per esempio un file che cancella lo schermo, o lo riempie di righe colorate, pronto per essere riciclato in applicazioni più personali (si fa per dire!).

L'importante, una volta riposto il floppy in mezzo agli altri, è ricordare di possedere quel determinato file, ed eventualmente dove andarlo a cercare.

Ed eccoci al punto cruciale: la necessità di tenere un archivio di tutti i nostri programmi, anche quelli che non appaiono sulle etichette dei floppy.

Tra l'altro, senza disporre di una lista adeguata, non è certo pensabile di accedere a scambi... più o meno interurbani.

BASIC E AMIGADOS

L'uso di un buon Database è il modo più ovvio per affrontare il problema ma, pur riempiendo di finestre lo schermo del nostro monitor (benedetto multitasking!), il compito risulterebbe in ogni caso abbastanza faticoso.

Sarebbe molto più semplice, e rapido, infilare ogni dischetto da archiviare nel drive e, scorrendone la directory, selezionare quali nomi di files prendere in considerazione e quali no.

Amiga Basic, se vogliamo ottenere qualcosa di simile, non può esserci direttamente di aiuto con i suoi comandi, che prevedono solo un misero "Files" per visionare la directory corrente, eventualmente stabilita tramite l'istruzione "Chdir".

Qualcosa di più, apparentemente, si può ottenere con il Cli, che prevede la possibilità di redirezionare l'output.

Per essere più chiari, una volta aperta una finestra Cli, con un comando tipo...

List >ram:nomefile

...da rileggere, eventualmente, con...
Type ram:nomefile

...si otterrà un file su Ram Disk (di nome

"nomefile") contenente l'elenco dei programmi nella directory corrente, ma, ovviamente, senza alcuna possibilità di scelta su quali ammettere e quali no nella lista.

Tuttavia esiste una via di mezzo che può fare al caso nostro.

Se, per esempio, da CLI si impartisce...

Dir >ram:buffer opt a

...viene generato il file di nome "buffer", anche in questo caso memorizzato nella Ram Disk, che includerà i nomi di tutti i programmi e delle directory presenti nel dischetto, come può essere verificato digitando (sempre da Cli):

Type Ram:buffer

Operando poi in Basic, sarà (quasi) facile manipolare il contenuto del file buffer per "impadronirsi" delle sue singole unità descrittive, utilizzabili nel modo che più ci aggrada: stampa su carta, archiviazione, e chi più ne ha più ne metta.

Vediamolo più chiaramente nella pratica, realizzando una impalcatura di base (solo software, niente mattoni!), comunque già operativa, sulla quale potrete poi sbizzarrirvi con i più disparati interventi.

QUALCHE PREPARATIVO

Il programma in questione lo si può vedere listato in queste pagine, ed il primo passo consiste ovviamente nel copiarlo dopo aver caricato Amiga Basic oppure, se si ha una certa pratica, utilizzando un word processor.

In quest'ultimo caso, com'è noto, il file va poi salvato in formato Ascii, scegliendo l'opzione adatta allo scopo (text, ascii, ecc.).

Nel dischetto in cui sarà memorizzato il programma, è consigliabile sia presente anche AmigaBasic, per facilitarne l'utilizzo.

Inoltre, obbligatoriamente, il sistema dovrà poter accedere ai comandi del dos "Run" e "Dir" (più "Assign" se non si dispone del secondo drive), nonchè al file "Dos.bmap".

I primi due (o tre), è arcinoto, sono presenti nella directory "C" del disco Workbench (1.2 oppure 1.3), mentre "Dos.bmap" è prelevabile dalla directory "BasicDemos" del dischetto Extras (quello che contiene il Basic, per intenderci), e va copiato nella directory "Libs" del disco corrente.

Per copiare sul dischetto di lavoro il file "Run", ad esempio, dovrete digitare...

Copy df0:c/run df1:

...dopo aver inserito il disco di workbench nel drive df0. Se disponete di un solo drive è invece necessario modificare la sintassi:

Copy c/run destinazione:

...in cui "destinazione" è il nome del dischetto di lavoro (attenzione al doppio punto finale).

Allo stesso modo, avendo assegnato il nome "extras1.3" (senza spazi tra i caratteri!) al disco origine e "destinazione a quello finale...

copy extras1.3:basicdemos/dos.bmap destinazione:

...che rimane valido anche se possedete due drive.

Per evitare complicazioni (in generale, non solo per questo programma, soprattutto se si è alle prime armi) si consiglia di preparare un dischetto con tutte le directory indispensabili (C, L, Libs, S, Devs, Fonts) copiandole dal floppy Workbench, aggiungervi Amiga Basic, copiare il file "dos.bmap" nella directory "Libs", ed infine installare il disco.

Così facendo, si disporrà di un ambiente di lavoro ottimale, che eviterà la solita marea di requester non sempre... amighevoli.

Ma torniamo al nostro listato, per una necessaria precisazione.

Nella sua prima parte, ben evidenziate, sono presenti un gruppo di righe non strettamente necessarie, se si dispone di due drive.

In caso contrario diventa obbligatorio copiarle, eliminando gli apici (=Rem) che le trasformano in righe di commento.

Tuttavia, anche con due drive, può risultare più comodo includere le righe di cui sopra.

Fate un po' voi, dopo avere sperimentato entrambi i modi.

POSTA AMIGA

COMPILATORE

□ **Vorrei sapere quale procedura bisogna seguire per ottenere dal True Basic un file direttamente eseguibile dal Dos, senza dover caricare l'interprete.**

(Federico Balbi - Spinea)

• Il Basic, sia esso il True, l'Amiga Basic, o l'ABasic, è un linguaggio tipicamente "interpretato", ovvero (lo dice la parola stessa), necessita di un interprete che legga dal testo le singole istruzioni e le esegua una per una ricorrendo ad apposite routine in linguaggio macchina.

L'Amiga, come quasi tutti i computer più evoluti, non dispone di una simile struttura software in ROM, per cui è necessario introdurla dall'esterno onde rendere eseguibile una routine scritta in Basic.

L'unico modo per rendere autonomo dall'interprete un programma, consiste nel compilarlo, in modo da avere un file già totalmente "tradotto", e quindi subito comprensibile per il sistema.

Mentre, però, per linguaggi come il C o lo stesso Assembly, un compilatore (o assemblatore) è parte integrante dello stesso linguaggio, per il Basic è di solito un'aggiunta "esterna", spesso disponibile solo grazie ad interventi esterni alla stessa ditta produttrice dell'interprete.

Non siamo a conoscenza dell'esistenza di compilatori per il True Basic (la sua opzione "compile" ha un significato diverso), mentre per l'AmigaBasic è in circolazione un AC Basic Compiler, che, oltre a velocizzare l'esecuzione del programma Basic, lo rende indipendente dall'interprete.

Anche in quest'ultimo caso, comunque, nel dischetto contenente il programma compilato dovranno essere presenti particolari files del compilatore.

CHE ORE SONO?

□ **Come si fa a regolare l'orologio dell'Amiga 500?**

(Stefano C. - Vacallo)

• Il modo più semplice è quello di ricorrere al programma "Preferences", presente nel dischetto Workbench, per settare a suon di "clickate" la data e l'ora del sistema.

Qualora si disponga della versione 1.3 è sufficiente uscire dalle Preferences tramite l'opzione Save, per ottenere la regolazione anche dell'orologio in tempo reale contenuto nell'eventuale espansione di memoria.

Nel caso in cui non si voglia modificare il file presente su Workbench, è necessario attivare la tacca di protezione ed ignorare (cliccando su Cancel) l'inevitabile richiesta di rimuovere la tacchetta; l'orologio in tampone dell'espansione verrà aggiornato egualmente.

AZIONE!

Ultimate le operazioni di copiatura non resta che scegliere l'opzione Start dal menu Run, per collaudare il programma.

Dopo un breve intervallo, necessario al sistema per accedere alla Dos.library (ne riparleremo tra breve), apparirà un nuovo schermo e la richiesta di inserire in df0: il disco da archiviare.

Se si possiedono due drive, e non si sono copiate le 11 righe evidenziate nel listato, sarà necessario spostare in df1: il disco con gli assegnamenti correnti, in caso contrario ci si limiti ad obbedire.

Se qualche finestra di sistema (requester) dovesse in questa fase far scomparire lo schermo di lavoro, niente paura: riapparirà presto, e, comunque, basta tirar giù la barra del workbench per essere certi che c'è ancora.

Compiuta l'operazione, si attenda che la spia del drive si spenga, dopodichè si prema Return.

Il drive interno si metterà in azione per un po', dopodichè apparirà sul video il nome della prima "entry" della directory, che potrà essere tanto il nome di un file quanto quello di una directory.

Sarà anche presente un breve menu che consente di scegliere, a seconda del tasto che si preme (la lettera evidenziata in rosso), se ignorare il file, se archiviarne il nome, o se modificarlo prima che venga archiviato.

In quest'ultimo caso, si aprirà una finestra (spostabile attraverso la barra del titolo) di input, che consentirà di digitarne il nuovo nome (seguito da Return!).

I file selezionati per l'archiviazione assumeranno uno sfondo arancione, mentre quelli ignorati manterranno le attribuzioni di colore standard, sempre che non abbiate modificato la palette di default attraverso le preferences.

Conclusa la scansione di tutti i files e directory, si potrà passare ad altri dischetti, mentre per uscire dal programma si dovrà pressare il tasto destro del mouse, e posizionarsi nell'unica opzione presente nel menu così accessibile.

Che cosa ne abbiamo ottenuto?
Semplice: un file di nome "Dbas", presente nella Ram Disk (ma posizionabile altrove, come vedremo tra breve), composto da tutti i nomi dei files scelti per l'archiviazione, ognuno dei quali prelevabile come singola stringa.

Da Basic, quindi, basterà mandare in esecuzione qualche riga di questo tenore...

```
open "ram:dbas" for input as 1
while not eof(1)
 input #1, nome$
 print nome$
wend
close #1
```

...per ottenerne la visualizzazione, ma anche, con facili modifiche, per assegnare i nomi ad un array, e farne ciò che si crede.

Nulla vieta, ad esempio, di aggiungere una routine di ordinamento, e, perchè no, sfruttare il Sort già fornito dal Dos.

La base su cui operare, insomma, è già pronta, ma può essere resa ancora più aderente ai nostri desideri comprendendo come ci si è arrivati.

CACCIA AL FILE

Da un punto di vista generale, i compiti che il programma assolve possono così essere schematizzati:

- Crea, sfruttando il Dos, un file provvisorio in Ram Disk (di nome "buff1") contenente la directory globale del dischetto da vagliare.

- Estrae dal file buffer1 i nomi dei singoli componenti della directory.

- Li sottopone alla scelta dell'utente e, se accettati, li aggiunge in coda ad un nuovo file, sempre su Ram Disk, di nome "dbas".

Tutto qui?
Ebbene sì, ma di materia da chiarire ce n'è più che abbastanza.

Cominciamo subito col primo punto, forse il più ostico.

In pratica, si tratta di inviare al dos il comando...

```
Dir >ram:buff1 opt a
```

...come se ci si trovasse in ambiente Cli.

Per far ciò è necessario ricorrere ad una funzione del Dos, chiamata Execute, che

---

Nella versione 1.2, anche dopo il Save Preferences, quest'ultimo non viene modificato, per cui si rende necessario aprire una finestra CLI, quindi impartire il comando "Setclock Opt Save", che trasferisce all'orologio hardware (se presente) l'ora e la data di sistema.

Con "Setclock opt load", si effettua l'operazione inversa, ovvero viene impostata come data ed ora corrente quella dell'orologio dell'espansione, ed è conveniente usare questa istruzione all'interno della startup-sequence del nostro disco di lavoro.

La prima regolazione (quella dell'orologio di sistema), oltre che dalle Preferences, può essere effettuata da Cli adoperando il comando Date, secondo la sintassi...

```
Date GG-MMM-AA HH:MM:SS
```

Con la versione 1.2, è obbligatorio far precedere da uno zero gli eventuali dati di una cifra (per esempio 08, e non 8), mentre la versione 1.3 accetta anche singoli caratteri numerici.

MONITOR

☐ **Quali sono le caratteristiche del monitor 2080 della Commodore?**
(Andrea Paolini - Pomezia)

• Anzitutto sarebbe più esatto dire "quali erano", visto che il modello 2080 sembra destinato ad un precoce pensionamento in favore di nuovi monitor dalle prestazioni più competitive.

Con il monitor 2080 veniva di molto ridotto, ma non completamente eliminato, il problema dello sfarfallio dell'immagine adoperando il modo interlacciato, ovvero la grafica con risoluzione 640 (o 320)x 512.

Il tutto, tra l'altro, ad un prezzo decisamente contenuto, se paragonato a quello di monitor professionali ad alta persistenza tipo il Nec (L. 1.800.000 circa).

Se, come mi sembra di capire dalla lettera, possiedi l'Amiga 2000, dai un'occhiata al numero 61 della rivista (pag. 70): potrai ricavarne qualche utile indicazione.

STAMPANTI

☐ **Vorrei procurarmi una stampante ad aghi usata. Posso collegarne al mio Amiga una qualsiasi, purchè possieda un connettore di tipo Centronics?**
(Claudio Bernardi - Sanremo)

• In teoria sì, ma è sempre meglio accertarsi che il modello di stampante da acquistare sia compreso tra quelli elencati nella directory Devs/Printers del dischetto Workbench, o comunque che sia "compatibile" con uno dei suddetti modelli.

La maggior parte delle stampanti in commercio, per esempio, implementano quello che è diventato quasi uno standard, l'emulazione Epson: se presente, anche un modello non citato tra le Printers del Workbench può andar bene.

Se, comunque, si teme di commettere errori di valutazione, c'è sempre una certa Commodore che, guarda caso, produce anche delle stampanti per Amiga...

potete vedere in azione nell'ultima subroutine del listato.

L'argomento, in effetti, richiederebbe un approfondimento ben al di là delle cognizioni legate al Basic, per cui limitiamoci qui a precisare che, inviando ad "Execute" una stringa contenente il comando Cli, o meglio il suo indirizzo effettivo in memoria (tramite Sadd), nonchè i parametri di input (0) ed output (pntr&, ovvero il puntatore alla stringa "ram:buff1"), il comando verrà eseguito.

Ovviamente, dovrà prima essere aperta la libreria relativa (vedi prime tre righe del listato), ed è per questo che si rende obbligatoria la presenza del file Dos.bmap nella directory Libs.

Più semplicemente, quando nel nostro listato appare una funzione...

dos x$, y$

...verrà eseguito il comando Cli identificato da X$, direzionato verso il device di output specificato da Y$.

Entrambe le stringhe, per una caratteristica propria del Dos, devono avere un chr$(0) come ultimo carattere.

Ridefinendo ogni volta x$ ed y$ nelle righe riservate a chi adopera solo il drive interno, vengono anche inviati al Dos tutti i comandi necessari a copiare in Ram Disk i vari files utili al programma, nonchè a modificare gli assegnamenti di C: e libs:.

Per la cronaca, il comando Run è indispensabile per il corretto funzionamento di Execute.

Il tutto non è poi così semplice da capire, ma, ahinoi, Amiga non è proprio un giocattolo...

Veniamo a cose più semplici, e soprattutto più "Basiche", tralasciando quanto facilmente interpretabile con l'aiuto(!) del solo manuale.

Il programma, sempre nelle prime righe, apre un file "ram:dbas" con la specifica Append; e quindi, se questo ancora non esiste, viene creato, altrimenti tutti gli output diretti verso questo file (con Print #37 verranno semplicemente accodati al suo precedente contenuto.

Sostituendo (per esempio) il nome del file con "df1:dbas", troveremo il file-archivio già bell'e pronto sul disco inserito nel drive 1.

L'istruzione On Break, poi, indirizzata verso un semplice Return, si limita ad impedire l'arresto del programma se non attraverso il Menu della barra-titolo, che obbliga al passaggio attraverso la routine di nome "uscita".

Questo per evitare che i file aperti rimangano tali, come pure lo schermo e la finestra creati ad hoc per il nostro miniarchivio (il Guru è sempre in agguato...).

Un altro punto da chiarire, riguarda le procedure seguite per determinare i singoli elementi stringa ricavati dal file buffer1, per essere poi trattati.

Il comando Dir, così come quando viene impartito da Cli, fornisce un output con i nomi dei file quasi sempre disposti a coppie per ogni riga.

Quindi, richiamando con Input #1 (il nostro file aperto in lettura) ogni singola stringa (entry$), questa potrà essere composta da un solo nome, o da due separati da un numero di spazi non sempre uguale.

Ecco allora che, alla label "legge1", viene utilizzato Instr$ per ricercare la posizione dei due spazi chr$(32) + chr$(32), che segnalano la fine del nome (volendo essere più sicuri, possono essere aggiunti altri spazi).

Se questi non vengono trovati, la variabile associata a Instr$ restituirà zero, indicando che non è presente un secondo nome di file.

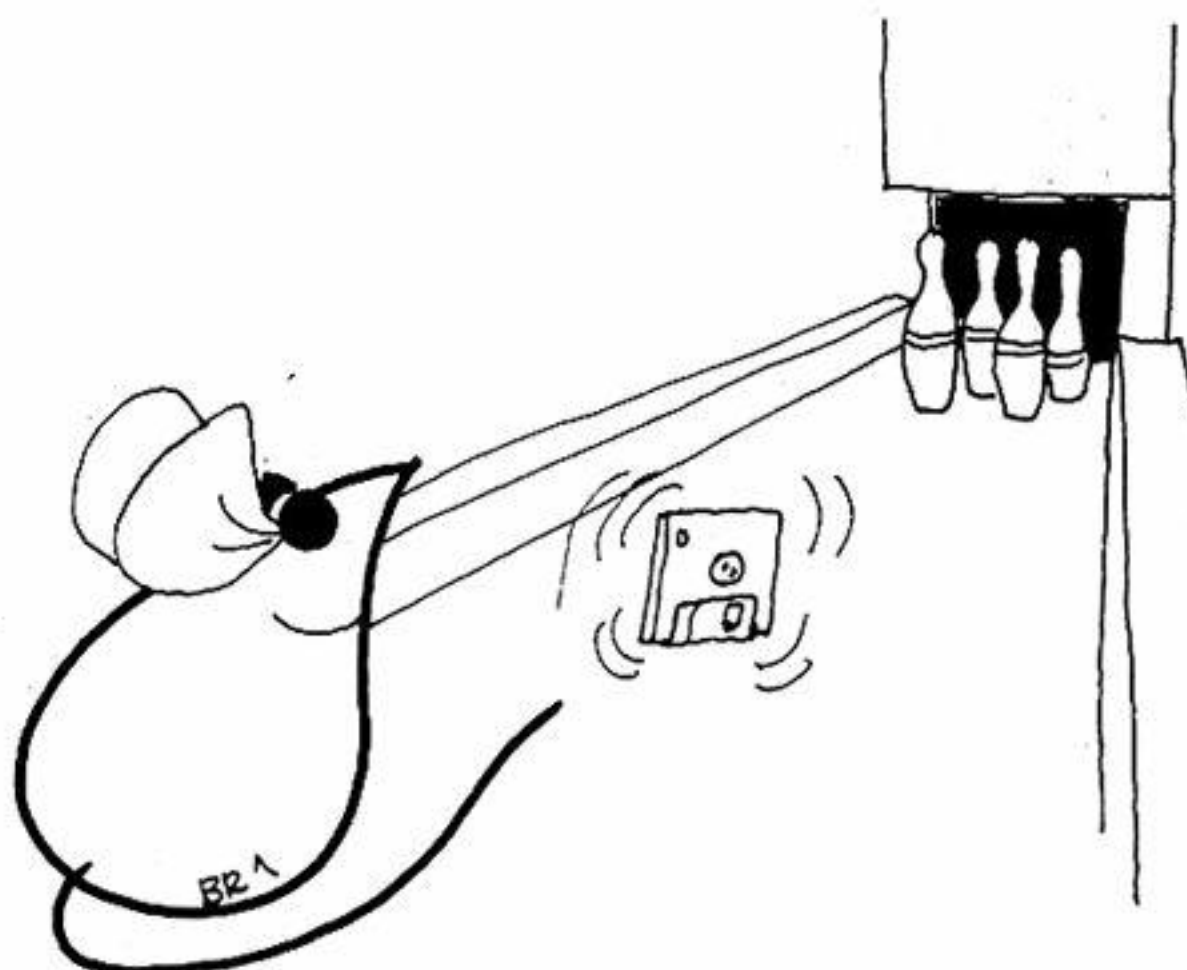
La stringa utile (entr2$), viene poi passata alla routine di scelta, che non merita particolari commenti.

Per l'eventuale secondo nome di file, si controlla invece, carattere per carattere, la stringa principale da destra verso sinistra (routine legge2), fino ad incontrare i fatidici due spazi.

Anche qui la stringa ricavata viene poi inviata alla routine di scelta.

Chiaro che, piuttosto che limitarsi a memorizzare i vari nomi nel file Dbas, questi potrebbero essere associati ad un array per essere ulteriormente processati, per esempio aggiungendo un menu che li ricolleghi alla posizione del dischetto, o iviati direttamente alla stampante con un Lprint.

Dopo quanto visto, in ogni caso, il più risulterà già fatto.

```
DECLARE FUNCTION xOpen& LIBRARY
DECLARE FUNCTION execute% LIBRARY
LIBRARY "dos.library"
OPEN"ram:dbas"FOR APPEND AS 3
SCREEN 1, 640, 256, 2, 2
WINDOW 2,"DiskBase",,23,1

'------------------------------------------
'  CHI NON POSSIEDE IL SECONDO DRIVE DEVE
'       AGGIUNGERE OBBLIGATORIAMENTE
'          LE 11 RIGHE CHE SEGUONO
'     (senza gli apici ad inizio riga!)
'------------------------------------------
'b$=CHR$(0):y$="nil:"+b$:z$="copy df0:"
'x$=z$+"libs/dos.bmap to ram:dos.bmap"+b$
'dos x$,y$
'x$=z$+"c/dir to ram:dir"+b$
'dos x$,y$
'x$=z$+"c/run to ram:run"+b$
'dos x$,y$
```

```
'x$="assign libs: ram:"+b$
'dos x$,y$
'x$="assign c: ram:"+b$
'dos x$,y$
'--------------------------------------

ON BREAK GOSUB ritorno
BREAK ON
MENU 1,0,1,"MENU"
MENU 1,1,1,"USCITA DAL PROGRAMMA"
MENU 2,0,0,""
MENU 3,0,0,""
MENU 4,0,0,""
ON MENU GOSUB uscita
MENU ON
FOR x=1 TO 35
 spazi35$=spazi35$+CHR$(32)
 IF x=7 THEN spazi7$=spazi35$
NEXT

start:
pag=0
CLS:PRINT "Inserisci disco da archiviare"
PRINT " in df0: e premi RETURN"
Loop:
a$=INKEY$
IF a$<>CHR$(13) THEN Loop
CLS:PRINT "attendi un momento..."
'----------------------------------------
x$="dir df0: opt a"+CHR$(0):REM comando cli
y$="ram:buff1"+CHR$(0):REM file buffer
dos x$, y$ :REM esegue comando cli
'----------------------------------------
CLS
OPEN "ram:buff1" FOR INPUT AS 1
WHILE NOT EOF(1)
'-------- PRELEVA PRIMO NOME FILE ---------
legge1:
 INPUT #1, entry$
 entr=LEN(entry$)
 y$=CHR$(32)+CHR$(32)
 x=INSTR(entry$,y$)
 IF x<>0 THEN
   entr2$=LEFT$(entry$,x-1)
 ELSE
     entr2$=entry$
     flag=1
 END IF
GOSUB scelta
entr2$=""
'--- CONTROLLA SE NOME 2 NON PRESENTE ---
IF flag=1 THEN
  flag=0
  GOTO fineciclo
END IF
' ------- PRELEVA SECONDO NOME FILE ------
legge2:
 x$=MID$(entry$,entr,1)
 IF x$ <>CHR$(32) THEN
   entr2$=x$+entr2$
   entr=entr-1
   GOTO legge2
 ELSE
   x$=MID$(entry$,entr-2,1)
   IF x$<>CHR$(32) THEN
     entr2$=CHR$(32)+entr2$
     entr=entr-1
     GOTO legge2
   END IF
 END IF
GOSUB scelta
entr2$=""
fineciclo:
WEND
CLOSE #1:GOTO start

'==========   subroutines   ============

scelta:
WINDOW 2
pag=pag+1
'------------ EVITA SCROLL -------------
 IF pag=27 THEN
  pag=0
  CLS
 END IF
'------ STAMPA MENU DI SCELTA -----------
x=CSRLIN : y=POS(0)
COLOR 1, 2
PRINT entr2$+spazi7$;
COLOR 1,3:PRINT "A";:COLOR 1,2
PRINT "rchivia - ";:COLOR 1,3
PRINT "M";:COLOR 1,2:PRINT "odifica - ";
COLOR 1,3:PRINT "I";:COLOR 1,2
PRINT "gnora "
'---------- ESEGUE SCELTA ---------------
risposta:
a$=INKEY$
IF a$<>"" THEN
a$=UCASE$(a$)
IF a$="A" THEN archivia
 IF a$ ="I" THEN
   LOCATE x, y
   COLOR 1,0
   PRINT entr2$+spazi7$+spazi35$
   RETURN
 END IF
 IF a$="M" THEN
   a2$="NUOVO NOME"
   WINDOW 3,a2$,(10,10)-(310,20),6,1
   INPUT "",entr2$
   WINDOW CLOSE 3
   GOTO archivia
 END IF
END IF
GOTO risposta
'----------------------------------------
uscita:
CLOSE
WINDOW CLOSE 2
SCREEN CLOSE 1
LIBRARY CLOSE
MENU RESET
END
'----------------------------------------
archivia:
LOCATE x,y:COLOR 1,3
PRINT entr2$;
COLOR 1,0
PRINT spazi7$+spazi35$
PRINT #3, entr2$

ritorno:
RETURN
'---------- ESEGUE COMANDI DOS ----------
 SUB dos (istr$, file$) STATIC
   pntr&=xOpen& (SADD(file$),1006)
   e% = execute% (SADD(istr$),0,pntr&)
   CALL xClose(pntr&)
 END SUB
```

IN SELLA AD AMIGA

Riprendiamo a parlare, dopo una lunga pausa, di alcune novità per Amiga e C/64; alcune belle, altre un po' meno...

di **Andrea Ciaramella**

SUPER HANG ON
(C/64 Amiga)

Direttamente dalla sala giochi arriva nelle nostre case l'attesissimo "SUPER HANG ON", il celeberrimo gioco arcade convertito per Amiga dalla Electric Dreams. Era già nota una versione del gioco per il Commodore 64, ma purtroppo si era rivelata un vero e proprio fallimento. Per Amiga, invece, grazie anche alle maggiori potenzialità tecniche dei computer a 16 bit, la conversione è risultata quanto di più simile si possa pensare all'originale arcade.

In questo gioco correte a bordo della vostra motocicletta attraverso America, Africa, Russia ed Europa per raggiungere i traguardi prefissati (chiamati "check") in un intervallo di tempo stabilito. Raggiunto ogni check, il tempo rimasto a disposizione verrà automaticamente incrementato e potrete così continuare a giocare sino al termine dell'intero circuito o allo scadere del tempo. Il gioco offre quattro differenti livelli, dal beginner (6 checks) all'expert (18 checks).

Dal punto di vista grafico il gioco è da considerare una vera perla: i fondali sono rifiniti con grande cura e lo scrolling è perfetto. Per quanto riguarda il sonoro, la "colonna sonora" è identica a quella della versione coin op ed il rumore del motore risulta incredibilmente realistico.

La motocicletta può essere comandata tramite joystick, mouse oppure tastiera; se si usa il mouse si può definire la sensibilità di movimento.

Il gioco non necessita di alcuna espansione anche se, possedendola, il computer non deve effettuare alcun caricamento durante i cambi di livello.

RING SIDE
(Amiga)

Sicuramente la maggior parte di noi ha desiderato fare una vita come quella di ROCKY BALBOA.

Con RING SIDE, l'ultimo prodotto della EAS, emulerete il vostro beniamino. Nel gioco dovrete dapprima "costruire" un pugile per farlo arrivare sino al vertice della graduatoria mondiale.

All'inizio del gioco il computer chiede il nome del pugile, il suo peso e l'età; in seguito vengono definiti i parametri di gioco, tra cui il numero e la durata di ciascun round, il livello di difficoltà e così via. Si provvede, quindi, a scegliere l'avversario contro cui combattere ed attendere che il computer carichi il gioco vero e proprio.

Durante il caricamento una ragazza con abiti piuttosto succinti (che farà la gioia dei vostri occhi di pugile stanco) annuncia il numero del round che sta per iniziare.

Nella schermata di gioco sono sempre presenti: la vista frontale del ring (con i pugili che si spostano orizzontalmente), i volti dei due pugili, le barre di energia (al termine delle quali il boxeur verrà messo K.O.), i punti assegnati ad ogni giocatore ed il tempo che deve trascorrere fino al termine del round.

La grafica è piuttosto buona, anche se non sfrutta al meglio le potenzialità dell'Amiga. La giocabilità, inoltre, è scarsa e ci si stanca velocemente. Durante l'intervallo, tra un round e l'altro, vi trovate nei panni di un "secondo" che ha il compito di massaggiare il proprio pugile, incrementando, così, il punteggio.

Ring Side è un videogioco che può piacere agli appassionati di boxe; tutti gli altri possono perderlo:

LIVE AND LET DIE
(Amiga)

Nel gioco vi calate nei panni di James Bond, il famoso agente 007, il quale, a bordo di un sofisticato motoscafo, è incaricato di scovare, e distruggere, la fabbrica di droga di Mr. Big a New Orleans.

Il videogioco è tratto dalla scena di inseguimento sui motoscafi del film omonimo, trasmesso di recente alla TV. All'inizio del gioco si può scegliere tra quattro diverse missioni: la prima si riferisce ad un allenamento al polo nord, la seconda nel deserto del Sahara; nella terza bisogna colpire il maggior numero possibile di bersagli fissi.

L'ultima missione è quella più importante: si parte per New Orleans.

L'accuratezza della grafica si spinge fino ai particolari più minuti; l'uso dei colori, poi, è veramente strabiliante; lo scrolling è molto buono e gli effetti sonori, (o la musica, a scelta) risultano davvero gradevoli. La giocabilità è soddisfacente e la varietà dei percorsi, anche se non molto estesa, conferisce al gioco quel tocco in più che lo rende completo.

Se siete smanettoni incalliti questo è il video gioco che fa per voi; se non lo siete, vi si presenta, con "Live and let die", l'occasione per diventarlo; inoltre, nei panni di 007, sarete attorniati da affascinanti fanciulle: cosa si può volere di più?

ZANY GOLF
(Amiga)

Avete mai giocato a minigolf? Zany golf si può definire una versione riveduta e corretta di questo sport. Le regole sono le stesse ma, in aggiunta, sono presenti buche che sarebbero impossibili o molto difficili da realizzare se non in un videogame.

Nel gioco (della Electronic Arts), infatti, venite catapultati in un immaginario percorso di minigolf ove sono presenti buche che si muovono da sole, palline che rotolano all'interno di enormi flipper ed altre simili stranezze.

Ogni buca è contraddistinta da un proprio "par" (numero di tiri consentiti per raggiungerla); l'obiettivo del gioco è di restare al di sotto del par, oppure di eguagliarlo. In caso contrario ogni tiro in più verrà sottratto da quelli inizialmente in dotazione; in seguito all'ultimo tiro apparirà sullo schermo il fatidico messaggio di "game over".

A facilitare il compito, altrimenti piuttosto arduo, vengono spesso assegnati tiri bonus. Ad esempio il giocatore, colpendo un fairy (pseudo insetto che si posa casualmente sul percorso) o terminando il percorso in tempi brevi, può disporre di altri tiri.

La grafica è ottimamente curata, carica di effetti molto simpatici, tra cui quello relativo alla bandierina che rientra nella buca appena la palla le si avvicina. Il sonoro è di ottima qualità e tutte le volte che una pallina cade nella buca viene generata una musica diversa.

Prima del successivo livello di gioco sullo schermo appare una rappresentazione, in piccolo, della buca ed alcune brevi istruzioni per completarla. Al gioco possono partecipare sino a quattro giocatori a turno, ognuno con una pallina di colore diverso.

Vince chi completa il maggior numero di buche usando il minor numero di colpi.

Il videogioco, che si gestisce mediante mouse, è veramente eccellente ma presenta il difetto di mettere a disposizione poche buche. Un data disk con nuovi percorsi non sarebbe una brutta idea: speriamo per il futuro!

WEC LE MANS

(Commodore 64)

Qual'è la più "dura" gara automobilistica? Ma come, non lo sapete? E' la famosa ventiquattr'ore di Le Mans.

La Electric Dreams ha prodotto una conversione dell'originale arcade che emula l'estenuante prova alla quale, naturalmente, tutti noi grandi piloti ci vogliamo sottoporre.

A differenza dalla gara "vera", quella su computer si snoda su quattro giri di circuito, divisi a loro volta in tre check, da raggiungere nel tempo stabilito.

Sicuramente vorrete sapere notizie sulla qualità della versione computerizzata; ad esser sinceri, non poteva essere peggiore.

Impiegherei troppo tempo per descrivere tutti i difetti del gioco (ma esiste qualche pregio?); vedremo quindi solo i principali.

La grafica dei fondali non è proprio delle migliori e non è paragonabile nemmeno lontanamente a ciò che si può ammirare nel noto "Super Cycle".

Gli sprites sono realizzati in stile Lego, colorati in maniera pessima e, se non bastasse, la macchina risponde ai comandi con lentezza esasperata. Gli effetti sonori del "rombo" del motore sembrano più adatti ad una zanzara affetta da laringite.

Lo scorrimento dello schermo è, a dir poco, ridicolo; di tanto in tanto si blocca senza alcuna ragione oppure si muove a scatti. L'effetto della velocità, elemento fondamentale di un gioco di corsa, è praticamente nullo e vi posso assicurare che non è, come si potrebbe insinuare, colpa del computer (ricordate lo scorrimento di "Buggy Boy"?).

Devo continuare?

Meglio di no. Se volete un vero gioco di corsa, lasciate perdere "Wec Le Mans": perfino il vecchio "Pole Position" lo batte ai punti.

TV SPORTS: FOOTBALL

(Amiga)

La Cinemaware ha colpito ancora con il suo ultimo prodotto: "Tv Football" entra trionfalmente nel campo delle simulazioni sportive.

Il gioco permette di partecipare al campionato di football americano NFL impersonando l'allenatore, il giocatore o lo spettatore televisivo.

Proprio così, se siete appassionati di football americano, e non vi potete permettere un'antenna parabolica per ricevere in diretta il campionato americano, il gioco di cui ci occupiamo è quello che stavate cercando. Vedrete, infatti, tutte le partite, superbowl compreso, esattamente come le vedrebbe un telespettatore degli USA.

All'inizio delle "trasmissioni" appare una sigla che annuncia la partita del giorno; in seguito viene trasmesso il notiziario sportivo con tanto di presentatore in giacca e cravatta.

Dopo l'introduzione, inizia lo scontro vero e proprio tra le squadre. Le fasi di gioco sono realizzate in maniera egregia e, a differenza di molti altri simulatori di football, tutti gli schemi sono eseguiti perfettamente dai giocatori computerizzati, particolare, questo, che vi fa realmente sentire i "padroni" del campo.

Dal menu principale potete scegliere se disputare una semplice "amichevole" oppure un intero campionato, altrimenti potete allenarvi simulando una partita o tentando di segnare dei tiri ben piazzati.Il gioco è pieno di scene puramente televisive, come quella del collegamento in diretta con gli spogliatoi oppure le saltuarie inquadrature del pubblico.

Tuttavia potrete eliminarle, abbreviando la durata dell'incontro. Tv Football si mantiene sui livelli standard della Cinemaware, conquistando il titolo di miglior simulazione di football americano per Amiga. Cosa state aspettando? Se fossi in voi correrei subito ad acquistarne una copia, non farlo è a vostro rischio e pericolo.

DRAGON NINJA

(Commodore 64)

Siete pronti? Vi siete scaldati i muscoli? Nella prossima missione, infatti, avrete molto bisogno di utilizzare i muscoli, più che il cervello.

Vi è stato assegnato il compito di distruggere una banda di fastidiosissimi ninja, accompagnati da gentili signore, che cercheranno di uccidervi. Enormi bestioni, che appaiono al termine di ogni livello, vi attaccano con armi pericolose.

Il gioco si svolge in diversi luoghi: fognature, foresta alquanto buia, città, tetto di un TIR ed altre località di vario tipo.

Per sopravvivere alla difficile situazione avrete a disposizione tre vite ed una quantità di tempo variabile a seconda del livello in cui vi trovate.

Scaduto il termine (o esaurite le vite) la missione avrà termine ed il vostro buon nome di ammazza-ninja sarà definitivamente infangato.

Sotto un punto di vista tecnico, il gioco risulta ottimamente realizzato. Ad esempio lo scrolling dei livelli del TIR e del treno è perfetto, la grafica è molto buona e gli effetti sonori sono nella norma.

La conversione dall'originale arcade è buona, con un'unica pecca: non esiste l'opzione per due giocatori che avrebbe permesso al gioco di diventare veramente una riproduzione fedele della macchina da bar.

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000

Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin AutoconfigTM - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000

Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000

Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB + sync -

Floppy Disk Driver A 1010 - L. 335.000

Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000

Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000

Scheda Janus XT+Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000

Scheda Janus AT+Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000

Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000

Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000

Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000

Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000

Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000

Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000

Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000

Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000

Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

A Scart - L. 28.000

Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 615.000

Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000

Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 8.930.000

Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3 2.1 - Interprete GW-Basic

PC60/40C - L. 9.180.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000

Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM: 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000

Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000

Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000

PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.965.000

Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 2.300.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.715.000

Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC20-IIIC - L. 3.050.000
Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000
Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000
Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128,64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000
Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000
Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000
Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D
1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64
Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64
Permette il collegamento a Videotel, P.G.E. e banche dati - **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000
Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000
Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000
Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000
Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000
Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000
Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000
Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000
Nastro per stampante

Stampante MPS 1500C - L. 550.000
Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000
Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000
Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

I COMMODORE POINT

LOMBARDIA

Milano
- AL RISPARMIO - V.le Monza, 204
- BCS - Via Montegani, 11
- BRAHA ALBERTO
- Via Pier Capponi, 5
- E.D.S. - Corso Porta Ticinese, 4
- E.S.C. - Via Roggia Scagna, 7
- FAREF - Via A. Volta, 21
- FLOPPERIA - Viale Monte Nero, 31
- GBC - Via Cantoni, 7
- Via Petrella, 6
- GIGLIONI LAURA - Via D'Ovidio, 8
- GILGLIONI - Viale Luigi Sturzo, 45
- LOGITEK - Via Golgi, 60
- MARCI [illegible] - Via F.lli Bronzetti, 37
- MEL[illegible]IONI - Via P. Colletta, 37
- MESSAGGERIE MUSICALI
- Galleria del Corso, 2
- NEWEL - Via Mac Mahon, 75
- RIVOLA - Via Vitruvio, 43

Provincia di Milano
- IL CURSORE - Via Cavour, 35
- Novate Milanese
- REC ELGRA - Corso Milano, 118
- Bovisio Masciago
- F.LLI GALIMBERTI
- Via Nazionale dei Giovi, 28/36
- Barlassina
- GBC - Viale Matteotti, 66
- Cinisello Balsamo
- P. GIORGIO OSTELLARI
- Via Milano, 300 - Desio
- CASA DELLA MUSICA
- Via Indipendenza, 21
- Cologno Monzese
- PENATI - Via Verdi, 28/30
- Corbetta
- EMP - Viale Italia, 12 - Corsico
- CENTRO COMPUTER PANDOLFI
- Via Corridoni, 18 - Legnano
- COMPUTEAM - Via Vecellio, 41
- Lissone
- FUTURA - Via Solferino, 31 - Lodi
- M.B.M. - Corso Roma, 12 - Lodi
- L'AMICO DEL COMPUTER
- Viale Lombardia, 17
- Melegnano
- BIT 84 - Via Italia, 4 - Monza
- I.C.O. - Via dei Tigli, 14 - Opera
- NIWA HARD & SOFT
- Via Bruno Buozzi, 94
- Sesto S. Giovanni
- FERRARI LUIGI - Via Madre Cabrini
- Sant'Angelo Lodigiano
- IL COMPUTER SERVICE SHOP
- Via Padana Superiore, 197
- Vimodrone

Bergamo
- COMIF - Via Autolinee, 10
- CORDANI - Via dei Caniana, 8
- D.R.B. - Via Borgo Palazzo, 65
- NEW SYSTEMS - Via Paglia, 36

Provincia di Bergamo
- BERTULEZZI G. - Via Fantoni, 48
- Alzano Lombardo
- COMPUTER TEAM - Via Verdi, 1/B
- Carvico
- OTTICO ROVETTA
- Piazza Garibaldi, 6 - Lovere
- A.I.S. INTERNATIONAL
- Via San Carlo, 25

San Pellegrino Terme
- SISTHEMA - Via Roma, 45
- Sarnico
- COMPUTER POINT
- Via Lantieri, 52 - Sarnico
- AB INFORMATICA
- Statale Cremasca, 66
- Urgnano

Brescia
- COMPUTER CENTER - Via Cipro, 6
- INFORMATICA 2000
- Via Stazione, 16/B
- MASTER INFORMATICA
- Via F.lli Ugoni, 10/B
- VIGASIO MARIO - Port. Zanardelli, 3

Provincia di Brescia
- MISTER BIT - Via Mazzini, 70 -Breno
- CAVALLI PIETRO
- Via 10 Giornate, 14/B
- Castrezzato
- VIETTI GIUSEPPE - Via Milano, 1/B
- Chiari
- MEGABYTE - Piazza Maluezzi, 14
- Desenzano del Garda
- CO-RE - Via XXV Aprile, 136 - Flero
- BARESI RINO & C.
- Via XX Settembre, 7 - Ghedi
- INFO CAM - Via Provinciale, 38
- Gratacasolo

Como
- IL COMPUTER - Via Indipendenza, 90
- 2M ELETTRONICA - Via Sacco, 3

Provincia di Como
- ELTRONGROS - Via L. da Vinci, 54
- Barzanò
- EGA - Via Mazzini, 42
- Cassago Brianza
- DATA FOUND - Via A. Volta, 4 -Erba
- EGA - Via A. Moro, 17 - Galbiate
- RIGHI ELETTRONICA
- Via G. Leopardi, 26
- Olgiate Comasco
- CIMA ELETTRONICA
- Via Leonardo da Vinci, 7
- Lecco
- FUMAGALLI - Via Cairoli, 48 -Lecco

Cremona
- MONDO COMPUTER
- Via Giuseppina, 11/B
- PRISMA - Via Buoso sa Dovara, 8
- TELCO - Piazza Marconi, 2/A

Provincia di Cremona
- ELCOM - Via IV Novembre, 56/58
- Crema
- EUROELETTRONICA
- Via XX Settembre, 92/A - Crema

Mantova
- TUBALDO ELSO & C.
- Galleria Fermi, 7
- 32 BIT - Via Cesare Battisti, 14
- ELETTRONICA BASSO
- V.le Risorgimento, 69

Pavia
- POLIWARE - C.so C. Alberto, 76

Provincia di Pavia
- LOGICA INFORMATICA
- Via Monte grappa, 32 - Vigevano
- M. VISENTIN - C.so V. Emanuele, 76
- Vigevano

Sondrio
- CIPOLLA M. - Via Tremogge, 25

Provincia di Sondrio
- FOTONOVA - Via Valeriana, 1
- San Pietro di Berbenno

Varese
- DIMECO SISTEMI - Via Garibaldi
- IL CENTRO ELETTRONICO
- Via Morazzone, 2
- SUPERGAMES - Via Carrobbio, 13

Provincia di Varese
- BUSTO BIT - Via Gavinana, 17
- Busto Arsizio
- CRESPI G.&C. - V.le Lombardia, 59
- Castellanza
- COMPUTER SHOP
- Via A. da Brescia, 2 - Gallarate
- LIMA IMPORT-EXPORT
- (Grandi Magazzini Bossi)
- Via Clerici, 196 - Gerenzano
- J.A.C. - Via Matteotti, 38
- Sesto Calende

VALLE D'AOSTA

Aosta
- F.lli GATTI - Via Festaz, 75

PIEMONTE

Alessandria
- BIT MICRO - Via Mazzini, 104
- SERVIZI INFORMATICI
- Via Alessandro III, 47

Provincia di Alessandria
- S.G.E. ELETTRONICA
- Via Bandello, 19 - Tortona

Asti
- RECORD - Corso Alfieri, 166/3

Cuneo
- ROSSI COMPUTER - C.so Nizza, 42
- STUDIO Software - C.so Nizza, 49

Provincia di Cuneo
- PUNTO BIT - Corso Langhe, 26/C
- Alba
- SDI - Via Vittorio Emanuele, 250
- Bra
- ASCHIERI GIANFRANCO
- Corso Emanuele Filiberto, 6
- Fossano

Novara
- ELCOM - Corso Mazzini, 11
- PROGRAMMA 3 - V.le Buonarroti, 8
- PUNTO VIDEO
- Corso Risorgimento, 39/B

Provincia di Novara
- MIRCO POLACCO & C.
- Via Monte Zeda, 4 - Arona
- ALL COMPUTER
- Corso Garibaldi, 106
- Borgomanero
- MICROLOGIC - Via Giovanni XIII, 2
- Domodossola
- ELLIOT COMPUTER
- Via Don Minzoni, 32 - Intra

Torino
- ALBA ELETTRONICA
- Via C. Fossati, 5/P
- ALEX COMPUTER
- Corso Francia, 333/4
- COMPUTER HOME
- Via San Donato, 46/D
- COMPUTING NEWS
- Via Marco Polo, 40/E
- DE BUG - C.so V. Emanuele II, 22
- DESME UNIVERSAL
- Via San Secondo, 95
- F.D.S. - Via Borgaro, 86/D
- INFORMATICA ITALIA
- Corso Re Umberto, 128
- MT INFORMATICA
- Corso Giulio Cesare, 58
- MUSICS'S Shop - C.so Potenza, 177
- NEW BUSINESS COMPUTER
- Via Nizza, 45/F
- PLAY GAMES - Via C. Alberto, 39/A
- RADIO TV MIRAFIORI
- C.so Unione Sovietica, 381
- SMIT ELETTRONICA
- Via Bibiana, 83/B
- TELERITZ - Corso Traiano, 34

Provincia di Torino
- PAUL E CHICO VIDEOSOUND
- Via V. Emanuele, 52 - Chieri
- BIT INFORMATICA
- Via V. Emanuele, 154 - Ciriè
- HI-FI CLUB - Corso Francia, 92/C
- Collegno
- I.C.S. - Stradale Torino N. 73
- Ivrea
- BAS - Corso Roma, 47
- Moncalieri
- CERUTTI MAURO
- Corso Torino, 234 - Pinerolo
- EUREX Corso Indipendenza, 5
- Rivarolo C.se
- DIAM INFORMATICA
- C.so Francia, 146/Bis - Rivoli
- FULLINFORMATICA
- Via Vittorio Veneto, 25 - Rivoli
- GAMMA COMPUTER
- Via Cavour, 3 A/B
- Settimo Torinese

Vercelli
- ELETTROGAMMA - C.so Bormida, 27
- ELETTRONICA di BELLAMO A. & C.
- Strada Torino, 15

Provincia di Vercelli
- C.S.I. TEOREMA - Via Losana, 9
- Biella
- SIGEST - Via Bertodano, 8 - Biella
- REMONDINO Franco - Via Roma, 5
- Borgosesia
- FOTOSTUDIO TREVISAN
- Via XXV Aprile, 24/B
- Cossato
- STUDIO FOTOGRAFICO IMARISIO
- Piazza Martiri Libertà, 7 - Trino

LIGURIA

Genova

ABM COMPUTER
- Piazza De Ferrari, 24/R
- CENTRO ELETTRONICA
- Via Chiaravagna, 10/R
- Località Sestri Ponente
- COMMERCIALE SOTTORIPA
- Via Sottoripa, 115/117
- FOTOMONDIAL
- Via del Campo, 3-5-9-11-13/R
- LA NASCENTE - Via San Luca, 4/1
- RAPPR - EL - Via Boreoratti, 23/R

Imperia
- CASTELLINO - Via Belgrano, 44
- SASA COMPUTER
- Via Nazionale, 256

Provincia di Imperia
- CENTRO HI-FI VIDEO
- Via della Repubblica, 38 - Sanremo
- CASTELLINO - Via Genova, 48
- Ventimiglia

La Spezia
- CATTONI - Via Vitt. Veneto, 75
- I.L. ELETTRONICA
- Via Vitt. Veneto, 123

Provincia di La Spezia
- I.L. ELETTRONICA - Via Aurelia, 299
- Fornola di Vezzano

Savona
- CASTELLINO
- Corso Tardy e Benech, 101
- ATHENA
- Via Carissimo E. Crotti, 16/R

VENETO

Belluno
- UP TO DATE
- Via Vittorio Veneto, 43

Provincia di Belluno
- GUERRA COMPUTERS
- Viale Mazzini, 10/C - Feltre

Padova
• BIT SHOP - Via Cairoli, 11
• COMPUMANIA
- Riviera Tiso Camposanpiero, 37
• COMPUTER POINT - Via Roma, 63
• D.P.R. - V.le Lombardo, 4
• GIANFRANCO MARCATO
- Via Madonna della Salute, 51/53
• ZELLA ADELIO
- Piazza De Gasperi, 31/A
Provincia di Padova
• CAERT - Via Andorra, 11
- Zona Industriale Camin
Treviso
• BIT 2000 - Via Brandolini d'Adda, 14
Provincia di Treviso
• DE MARIN - Vai XX Settembre, 74
- Conegliano
• SIDESTREET - Via S. d'Acquisto, 8
- Montebelluna
• FALCON - Via Terraggio, 116
- Preganziol
Venezia
• TELERADIO FUGA
- San Marco, 3457
Provincia di Venezia
• GUERRA EGIDIO & C.
- Via Bissuola, 20/A - Mestre
• TREKILOWATT
- Via Torre Belfredo, 47 - Mestre
• REBEL - Via F. Crispi, 10
- San Donà di Piave
• GUERRA COMPUTERS
- Via Vizzotto, 29
- San Donà di Piave
• TELFERT - Via Chiesa, 1509
- Sottomarina
• RADIOCESTARO - Via Roma, 89
- Spinea
Verona
• CASA DELLA RADIO - Via Cairoli, 10
• TELESAT - Via Vasco de Gama, 8
Provincia di Verona
• FERRARIN - Via dei Massari, 10
- Legnago
Vicenza
• ELETTRONICA BISELLO
- Viale Trieste, 427/429
• SCLACHI Market - Via Cà Balbi, 139
Provincia di Vicenza
• GUERRA COMPUTERS
- Via Dell'Industria - Alte Ceccato
• SCIAVOTTO - Via Zanella, 21
- Cavazzale
• COMPUTER B. COSTO
- Via del Costo, 34 - Thiene
• ELETTROCASA - Via Roma, 67
- Trissino

FRIULI VENEZIA GIULIA
Gorizia
• E.C.O. ELETTRONICA
- Via F.lli Cossar, 23
Pordenone
• RIGO - Viale Cossetti, 5
Provincia di Pordenone
• MDT - Piazza Repubblica, 5
- Villanova di Prata
• BRUNO DA PIEVE
- Via Colombera, 17 - Porcia
Trieste
• AVANZO GIACOMO
- Piazza Cavana, 7
• COMPUTER SHOP - Via P. Reti, 6
• COMPUTIGI - Via XX Settembre, 51
• CTI - Via Pascoli, 4

Udine
• MOFERT 2 - Via Leopardi, 21
• R.T. SYSTEM - Via L. da Vinci, 99
Provincia di Udine
• IDRENO MATTIUSSI & C.
- Via Liciniana, 50 - Tavagnacco
Rovigo
• CLINICA DEL RASOIO
E DEL COMPUTER
- Via Fiume, 31/33

TRENTINO ALTO ADIGE
Bolzano
• C.M.B. ITALIA - Via Roma, 82
• MATTEUCCI PRESTIGE
- Via Museo, 54
Provincia di Bolzano
• ELECTRO TAPPEINER
- P.za Principale, 90 - Silandro
• RADIO MAIR - Via Centrale, 70
- Brunico
• ELECTRO RADIO HENDRICH
- Via delle Corse, 106 - Merano
Trento
• CRONST - Via Galilei, 25

EMILIA ROMAGNA
Piacenza
• COMPUTER Line - Via G. Carducci, 4
• DELTA COMPUTER
- Via Martiri della Resistenza, 15/G
• SOVER - Via IV Novembre, 60

TOSCANA
Arezzo
• DELTA SYSTEM - Via Piave, 13
Firenze
• ATEMA
- Via Bendetto Marcello, 1a/1b
• COOPERATIVA L.D.T. - Via Icaro, 9
• ELETTRONICA CENTOSTELLE
- Via Centostelle, 5/a
• HELP COMPUTER
- Via degli Artisti, 15/A
• PUNTO SOFT - Via Vagnetti, 17
• TELEINFORMATICA TOSCANA
- Via Bronzino, 36
Provincia di Firenze
• WAR GAMES
- Via Raffaello Sanzio, 126/A
- Empoli
• NEW E.V.M. COMPUTER
- Via degli Innocenti, 2
- Figline Valdarno
• COSCI F.LLI - Via Roma, 26 - Prato
• CENTRO INFORMATICA
- Via F.lli Cervi, 21/29 - Pontassieve
Grosseto
• COMPUTER SERVICE
- Piazza Ponchielli, 2
Livorno
• ETA BETA - Via San Francesco, 30
• FUTURA 2 - Via Cambini, 19
Provincia di Livorno
• ELETTRONICA ALESSI PAOLO
- Via Cimarosa, 1 - Piombino
Provincia di Lucca
• IL COMPUTER - V.le Colombo, 216
- Lido di Camaiore
• SANTI VITTORIO - Via Roma, 23
- San Romano Garfagnana
Massa
• EURO COMPUTER
- Piazza Bertagnini, 4
• FIRMWARE - Via Aurelia Ovest, 27

Provincia di Massa
• RADIO LUCONI - Via Roma, 24/B
- Carrara
Pisa
• C.H.S. - Via Carlo Cattaneo, 90/92
• ELECTRONIC SERVICE
- Via della Vecchia Tranvia, 10
• IT - LAB - Via Marche 8A/8B
Pistoia
• ELECTRONIC SHOP
- Via della Madonna, 49
• OFFICE DATA SERVICE
- Galleria Nazionale, 22
Provincia di Pistoia
• ZANNI & C. - Corso Roma, 45
- Montecatini Terme
Siena
• VIDEO MOVIE - Via Garibaldi, 17
Provincia di Siena
• ELECTRONIC Shop - Via A. Casini, 51
- Chianciano Terme
• ELETTRONICA
- Via di Gracciano nel Corso, 111
- Montepulciano

UMBRIA
Provincia di Perugia
• COMPUTER STUDIOS
- Via IV Novembre, 18/A
- Bastia Umbra

CAMPANIA
Provincia di Avellino
• FLIP FLOP - Via Appia, 68 -Atripalda
Benevento
• E.CO: INFORMATICA
- Via Pepicelli, 21/25
Caserta
• O.P.C. - Via G.M. Bosco, 24
Provincia di Caserta
• M.P. COMPUTER - Via Napoli, 30
- Maddaloni
Napoli
• BABY TOYS
- Via Cisterna dell'Olio, 5/Bis
• CASA MUSICALE RUGGIERO
- Piazza Garibaldi, 74
• CENTRO ELETTRONICO CAMPANO
- Via Epomeo, 121
• C.I.AN - Galleria Vanvitelli, 32
• DARVIN - Calata San Marco, 26
• ELETTRONICA RO.DA.LO.
- Via Epomeo, 216/B
• GIANCAR 2 - Piazza Garibaldi, 37
• GRUPPO BUSH
- Galleria Umberto I, 55
• ODORINO - Largo Lala, 22/A-B
• R 2 - Via F. Cilea, 285
• SPY - Via Fontana, 135
• TOP - Via S. Anna dei Lombardi, 12
• VIDEOFOTOMARKET
- Via S. Brigida, 19
Provincia di Napoli
• SPADARO - Via Romani, 93
- S. Anastasia
• TUFANO - S.S. Sannitica, 87 Km 7
- Casoria
• ELETTRONICA 2000
- Corso Durante, 40
- Frattamaggiore
• GATEWAY - Via Napoli, 68
- Mugnanò
• NUOVA INFORMATICA SHOP
- Via Libertà, 185/191 - Portici
• BASIC COMPUTER
- C.so Garibaldi, 34
- Pozzuoli

• FALCO ELETTRONICA
- Via Sarno, 100 - Striano
• TECNOTRE - Via P. Fusco, 1/F
- Torre Annunziata
Salerno
• COMPUTER MARKET
- C.so Vitt. Emanuele, 23
Provincia di Salerno
• KING COMPUTER - Via Olevano, 56
- Battipaglia
• DIMER POINT - Via C. Rosselli, 20
- Eboli

PUGLIA

Bari
• ARTEL - Via G. d'Orso, 9
• COMPUTER'S ARTS
- Viale Meucci, 12/B
Provincia di Bari
• F. FAGGELLA - Corso Garibaldi, 15
- Barletta
• G. FAGGELLA
- Via P. d'Aragona, 62/A - Barletta
• G. LONUZZO - Via Nizza, 21
- Castellana
Brindisi
• MARANGI & MICCOLI
- Via Prov. San Vito, 165
Provincia di Foggia
• IL DISCOBOLO - Via T. Solis, 15
- San Severo
Lecce
• BIT
- Via 95° Regg.to Fanteria, 87/89
Provincia di Lecce
• CEDOK INFORMATICA - Via Roma, 31
- Tricase
Matera
• GUADIANO ELECTRONICS
- Via Roma, 1
Taranto
• ELETTROJOLLY - Via de Cesare, 13
• TEA - Via Regina Elena, 101

CALABRIA

Catanzaro
• C. & G. COMPUTER - Via F. Acri, 28
• PAONE SAVERIO - Via F. Acri, 93/99
Provincia di Catanzaro
• COMPUTER HOUSE - Via Bologna
- Crotone
• OTTICA FOTO NELLO RUELLO
- C.so Vittorio Emanuele, 177
- Vibo Valentia
Cosenza
• SIRANGELO COMPUTER
- Via N. Parisio, 25
Provincia di Cosenza
• ELIGIO ANNICHIARICO & C.
- Via Roma, 21 - Castrovillari
• ALFA COMPUTER
- Via Nazionale, 341/A
- Corigliano Scalo
• ING. FUSTO SALVATORE
- Corso Nicotera, 99
- Lamezia Terme
Reggio Calabria
• CONTROL SYSTEM
- Via S. Francesco da Paola, 49/DE
• SYSTEM HOUSE
- Via Fiume ang. Palestino, 1
Provincia di Reggio Calabria
• COMPUTER SHOP
- Via Matteotti, 50/52 - Locri

N. 10 - L. 12.000

Commodore 64 Club

QUATTRO SPLENDIDI VIDEOGIOCHI, DUE MEGA-UTILITY E IN PIU'... TUTTI I PROGRAMMI DI COMMODORE COMPUTER CLUB N. 62

FANTASTICO

POINTER: TRASFORMA IL TUO 64 IN UN AMIGA E UTILIZZA IL JOYSTICK COME IL MOUSE

PUNKILLER: SFIDA AI GLADIATORI DEL 2000

é in edicola

IN OMAGGIO DA QUESTO NUMERO TUTTI I PROGRAMMI DI COMMODORE COMPUTER CLUB